MATLAB PDE Solving

Boundary Value Problems

Computational Design Laboratory
Department of Automotive Engineering
Hanyang University, Seoul, Korea

CONTENTS

- **Boundary Value Problems for PDE**

  - ✓ **Finite Difference Method: Elliptic equation**

LAPLACE EQUATION

– PDE → algebraic difference equation

$$\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}=0 \leftarrow \begin{cases}\dfrac{\partial^2 T}{\partial x^2}=\dfrac{T_{i+1,j}-2T_{i,j}+T_{i-1,j}}{\Delta x^2}\\ \dfrac{\partial^2 T}{\partial y^2}=\dfrac{T_{i,j+1}-2T_{i,j}+T_{i,j-1}}{\Delta y^2}\end{cases}$$

$$\rightarrow \frac{T_{i+1,j}-2T_{i,j}+T_{i-1,j}}{\Delta x^2}+\frac{T_{i,j+1}-2T_{i,j}+T_{i,j-1}}{\Delta y^2}=0$$

$$\xrightarrow{\Delta x=\Delta y} T_{i+1,j}+T_{i-1,j}+T_{i,j+1}+T_{i,j-1}-4T_{i,j}=0$$

apply boundary conditions (fixed/Dirihlet)

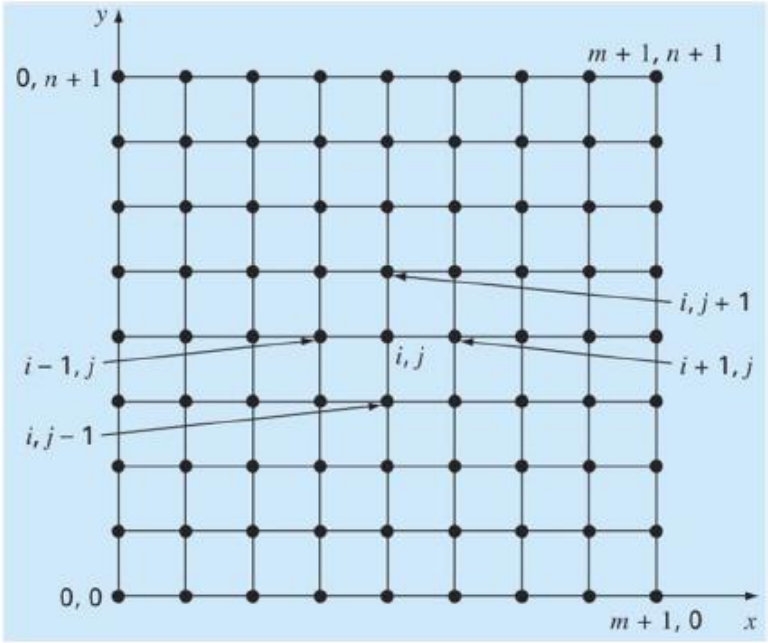
$$@(1,1): T_{21}+\underbrace{T_{01}}_{75°C}+T_{12}+\underbrace{T_{10}}_{0°C}-4T_{11}=0 \rightarrow 4T_{11}-T_{21}-T_{12}=75$$

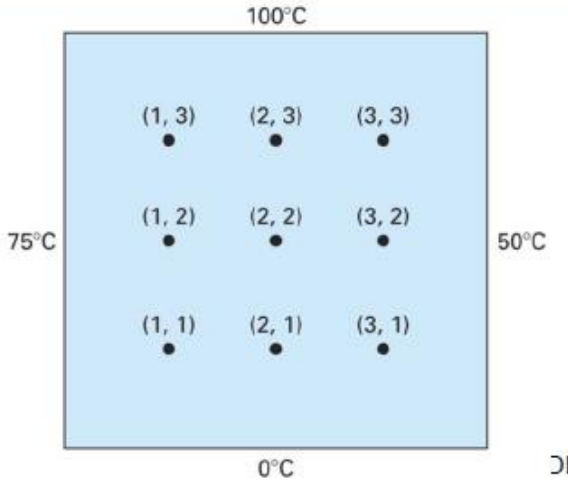
$$@(2,1): T_{31}+T_{11}+T_{22}+\underbrace{T_{20}}_{0°C}-4T_{21}=0 \rightarrow -T_{11}+4T_{21}-T_{31}-T_{22}=0$$

$$@(3,1): \underbrace{T_{41}}_{50°C}+T_{21}+T_{32}+\underbrace{T_{30}}_{0°C}-4T_{31}=0 \rightarrow -T_{21}+4T_{31}-T_{32}=50$$

$\vdots$

9 equations





$$[K_{ij}]\{T_i\}=\{f_i\}$$

$[K_{ij}]$: coefficient matrix, $\{T_i\}$: solution vector, $\{f_i\}$: force vector

MATLAB CODE

```
clc; clear all;
T_left = 75; T_right = 50; T_bottom = 0; T_upper = 100;
nx = 3; ny = 3;
T_numbering = [];
iter = 0;
i = [1:nx]';
j = ones(nx,1);
for k = 1:ny
    T_numbering = [T_numbering;i,j*k];
end
f = zeros(nx*ny,1);
for j = 1:ny
    for i = 1:nx
        T_numbering(:,3) = zeros(nx*ny,1);
        a = find(T_numbering(:,1) == i+1 & T_numbering(:,2) == j);
        b = find(T_numbering(:,1) == i-1 & T_numbering(:,2) == j);
        c = find(T_numbering(:,1) == i & T_numbering(:,2) == j+1);
        d = find(T_numbering(:,1) == i & T_numbering(:,2) == j-1);
        e = find(T_numbering(:,1) == i & T_numbering(:,2) == j);
        T_numbering(a,3) = -1;
        T_numbering(b,3) = -1;
        T_numbering(c,3) = -1;
        T_numbering(d,3) = -1;
        T_numbering(e,3) = 4;
        iter = iter + 1;
        K(iter,:) = T_numbering(:,3)';
        if length(a) == 0
            f(iter) = f(iter) + T_right;
        end
        if length(b) == 0
            f(iter) = f(iter) + T_left;
        end
        if length(c) == 0
            f(iter) = f(iter) + T_upper;
        end
        if length(d) == 0
            f(iter) = f(iter) + T_bottom;
        end
    end
end
Temp_temp1 = K\f;
iter = 0;
for i = ny:-1:1
    for j = 1:nx
        iter = iter + 1;
        Temp_temp2(i,j) = Temp_temp1(iter,1);
    end
end
Temp = zeros(ny+2,nx+2);
Temp(:,1) = T_left;
Temp(:,nx+2) = T_right;
Temp(1,:) = T_upper;
Temp(ny+2,:) = T_bottom;
Temp(2:ny+1,2:nx+1) = Temp_temp2;
x=[0:1:nx+1]; y=[ny+1:-1:0];
surf(x,y,Temp)
xlabel('x'); ylabel('y');zlabel('Temp'); colorbar
```

INITIALIZATION

```
clc; clear all;

T_left = 75; T_right = 50; T_bottom = 0; T_upper = 100;

nx = 3; ny = 3;

T_numbering = [];
iter = 0;

i = [1:nx]';
j = ones(nx,1);

for k = 1:ny
    T_numbering = [T_numbering;i,j*k];
end

f = zeros(nx*ny,1);
```

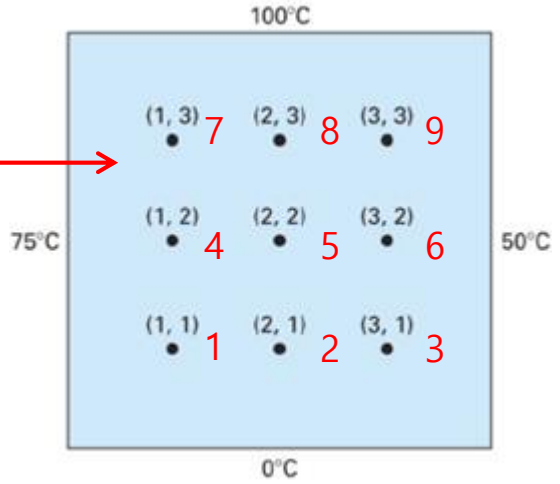
1 경계조건 선언

2 경계를 제외한 node 개수 선언

3 매트릭스를 구성하기 위한 정보를 저장

	1	2
1	1	1
2	2	1
3	3	1
4	1	2
5	2	2
6	3	2
7	1	3
8	2	3
9	3	3

numbering



COEFFICIENT MATRIX

```
for j = 1:ny
    for i = 1:nx
        T_numbering(:,3) = zeros(nx*ny,1);
        a = find(T_numbering(:,1) == i+1 & T_numbering(:,2) == j);
        b = find(T_numbering(:,1) == i-1 & T_numbering(:,2) == j);
        c = find(T_numbering(:,1) == i & T_numbering(:,2) == j+1);
        d = find(T_numbering(:,1) == i & T_numbering(:,2) == j-1);
        e = find(T_numbering(:,1) == i & T_numbering(:,2) == j);
        T_numbering(a,3) = -1;
        T_numbering(b,3) = -1;
        T_numbering(c,3) = -1;
        T_numbering(d,3) = -1;
        T_numbering(e,3) = 4;
        iter = iter + 1;

        K(iter,:) = T_numbering(:,3)';
```

1 K 매트릭스 생성

$$[K_{ij}]\{T_i\} = \{f_i\}$$

2 nx = 3, ny = 3 일 때의 매트릭스 요소 정보

numbering

	1	2
1	1	1
2	2	1
3	3	1
4	1	2
5	2	2
6	3	2
7	1	3
8	2	3
9	3	3

T <double>

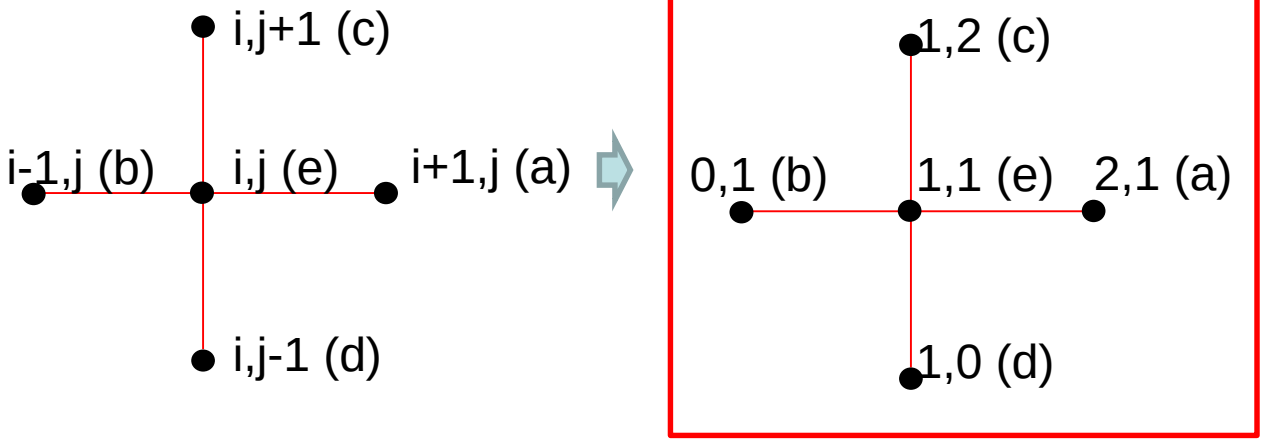
	1	2	3	4	5	6	7	8	9
1	4	-1	0	-1	0	0	0	0	0
2	-1	4	-1	0	-1	0	0	0	0
3	0	-1	4	0	0	-1	0	0	0
4	-1	0	0	4	-1	0	-1	0	0
5	0	-1	0	-1	4	-1	0	-1	0
6	0	0	-1	0	-1	4	0	0	-1
7	0	0	0	-1	0	0	4	-1	0
8	0	0	0	0	-1	0	-1	4	-1
9	0	0	0	0	0	-1	0	-1	4

COEFFICIENT MATRIX

```
a = find(T_numbering(:,1) == i+1 & T_numbering(:,2) == j);
b = find(T_numbering(:,1) == i-1 & T_numbering(:,2) == j);
c = find(T_numbering(:,1) == i & T_numbering(:,2) == j+1);
d = find(T_numbering(:,1) == i & T_numbering(:,2) == j-1);
e = find(T_numbering(:,1) == i & T_numbering(:,2) == j);
```

find 는 벡터 혹은 매트릭스 내 요소 중 같은 값을 갖는 요소의 위치정보를 저장하는 명령어

i = 1, j = 1 일 때 T_numbering 벡터의 위치 정보를 저장



		1	2	
e	1	1	1	a=2
a	2	2	1	b=[]
	3	3	1	
c	4	1	2	c=4
	5	2	2	d=[]
	6	3	2	
	7	1	3	e=1
	8	2	3	
	9	3	3	

COEFFICIENT MATRIX

```
T_numbering(a,3) = -1;
T_numbering(b,3) = -1;
T_numbering(c,3) = -1;
T_numbering(d,3) = -1;
T_numbering(e,3) = 4;
```

	1	2	3
1	1	1	4
2	2	1	-1
3	3	1	0
4	1	2	-1
5	2	2	0
6	3	2	0
7	1	3	0
8	2	3	0
9	3	3	0

1. 위치 정보를 이용하여 계수 값을 T_numbering 내에 입력
2. 입력된 3번째 벡터를 coefficient matrix 의 첫 번째 열로 저장

$$@(1,1): T_{21} + \underbrace{T_{01}}_{75°C} + T_{12} + \underbrace{T_{10}}_{0°C} - 4T_{11} = 0 \rightarrow 4T_{11} - T_{21} - T_{12} = 75$$

$$@(2,1): T_{31} + T_{11} + T_{22} + \underbrace{T_{20}}_{0°C} - 4T_{21} = 0 \rightarrow -T_{11} + 4T_{21} - T_{31} - T_{22} = 0$$

$$@(3,1): \underbrace{T_{41}}_{50°C} + T_{21} + T_{32} + \underbrace{T_{30}}_{0°C} - 4T_{31} = 0 \rightarrow -T_{21} + 4T_{31} - T_{32} = 50$$

$$\vdots$$

9 equations

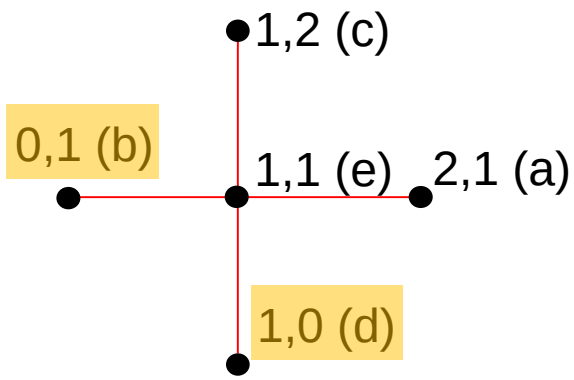
```
K(iter,:) = T_numbering(:,3)';
```

	1	2	3	4	5	6	7	8	9
1	4	-1	0	-1	0	0	0	0	0

$$\begin{bmatrix} k_{11} & k_{12} & \cdots & k_{12N} \\ k_{21} & k_{22} & & k_{22N} \\ \vdots & & \ddots & \\ k_{2N1} & k_{2N2} & & k_{2N2N} \end{bmatrix} \begin{bmatrix} u_1^{(1)} \\ u_2^{(1)} \\ \vdots \\ u_2^{(N)} \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_4 \end{bmatrix}$$

FORCE VECTOR

```
if length(a) == 0
    f(iter) = f(iter) + T_right;
end
if length(b) == 0
    f(iter) = f(iter) + T_left;
end
if length(c) == 0
    f(iter) = f(iter) + T_upper;
end
if length(d) == 0
    f(iter) = f(iter) + T_bottom;
end
end
end
```



경계조건이 입력될 부분

※ example

a=2
b=[]
c=4
d=[]
e=1

if length(a) == 0

If isempty(a)

0
1
0
1
0

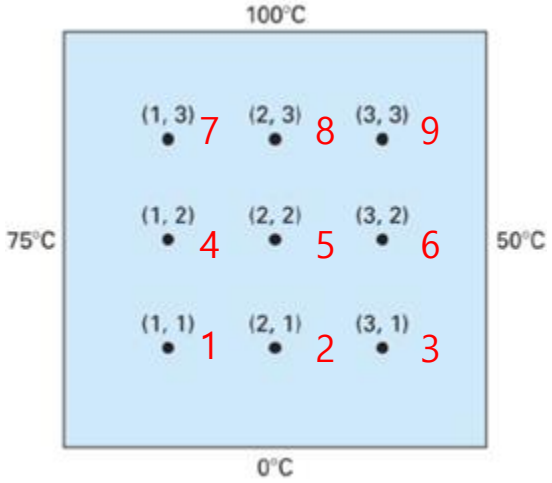
경계조건을 입력할 번호를 구분, length 명령어를 이용하여 값이 0 일 경우 경계에 존재하는 번호
(또는 isempty 함수사용)

a 가 경계일 경우 항상 오른쪽 경계
마찬가지 논리로 입력할 경계 위치를 결정

$$@(1,1): T_{21} + \underbrace{T_{01}}_{75°C} + T_{12} + \underbrace{T_{10}}_{0°C} - 4T_{11} = 0 \rightarrow 4T_{11} - T_{21} - T_{12} = 75$$

$$@(2,1): T_{31} + T_{11} + T_{22} + \underbrace{T_{20}}_{0°C} - 4T_{21} = 0 \rightarrow -T_{11} + 4T_{21} - T_{31} - T_{22} = 0$$

$$@(3,1): \underbrace{T_{41}}_{50°C} + T_{21} + T_{32} + \underbrace{T_{30}}_{0°C} - 4T_{31} = 0 \rightarrow -T_{21} + 4T_{31} - T_{32} = 50$$



	1
1	75
2	0
3	50
4	75
5	0
6	50
7	175
8	100
9	150

POST-PROCESSING

```
Temp_temp1 = K\f;
```

$$[K_{ij}]\{T_i\} = \{f_i\} \Rightarrow \{T_i\} = [K_{ij}]^{-1}\{f_i\}$$

1 solve

2 솔루션 매트릭스 재배열

재배열을 하는 것은 온도분포가 1열 벡터로 출력되기 때문

surf 명령어로 후처리를 할 예정이므로 매트릭스로 재배열이 필요

```
iter = 0;
for i = ny:-1:1
    for j = 1:nx
        iter = iter + 1;
        Temp_temp2(i,j) = Temp_temp1(iter,1);
    end
end
```

2

Temp

5x5 double

	1	2	3	4	5
1	100	100	100	100	100
2	75	78.5714	76.1161	69.6429	50
3	75	63.1696	56.2500	52.4554	50
4	75	42.8571	33.2589	33.9286	50
5	0	0	0	0	0

3 후처리

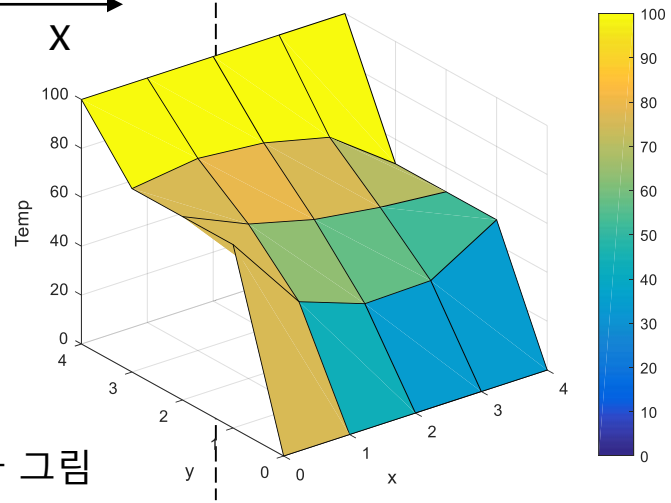
```
Temp = zeros(ny+2,nx+2);
Temp(:,1) = T_left;
Temp(:,nx+2) = T_right;
Temp(1,:) = T_upper;
Temp(ny+2,:) = T_bottom;
Temp(2:ny+1,2:nx+1) = Temp_temp2;
x=[0:1:nx+1]; y=[ny+1:-1:0];
surf(x,y,Temp)
xlabel('x'); ylabel('y');zlabel('Temp'); colorbar
```

y

0

x

3



결과 그림

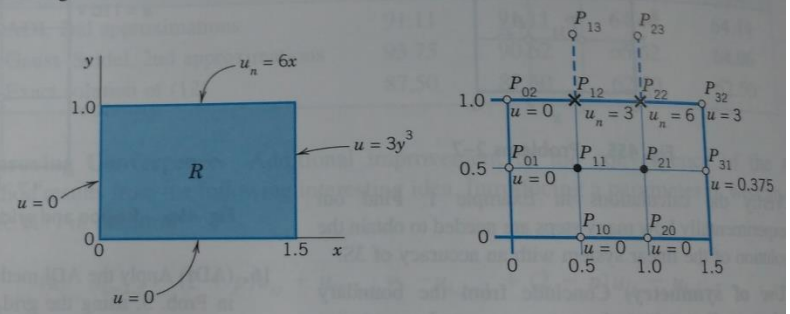
ASSIGNMENT

EXAMPLE 1 Mixed Boundary Value Problem for a Poisson Equation

Solve the mixed boundary value problem for the Poisson equation

$$\nabla^2 u = u_{xx} + u_{yy} = f(x, y) = 12xy$$

shown in Fig. 457a.



(a) Region R and boundary values (b) Grid ($h = 0.5$)

Fig. 457. Mixed boundary value problem in Example 1

(3)
$$\begin{bmatrix} -4 & 1 & 1 & 0 \\ 1 & -4 & 0 & 1 \\ 2 & 0 & -4 & 1 \\ 0 & 2 & 1 & -4 \end{bmatrix} \begin{bmatrix} u_{11} \\ u_{21} \\ u_{12} \\ u_{22} \end{bmatrix} = \begin{bmatrix} 0.75 \\ 1.125 \\ 1.5 - 3 \\ 0 - 6 \end{bmatrix} = \begin{bmatrix} 0.75 \\ 1.125 \\ -1.5 \\ -6 \end{bmatrix}.$$

(The entries 2 come from u_{13} and u_{23}, and so do -3 and -6 on the right). The solution of (3) (obtained by Gauss elimination) is as follows; the exact values of the problem are given in parentheses.

$u_{12} = 0.866$ (exact 1) $u_{22} = 1.812$ (exact 2)

$u_{11} = 0.077$ (exact 0.125) $u_{21} = 0.191$ (exact 0.25). ■

$$u_{xx} + u_{yy} = f(x, y) = 12xy$$

Dirichet B.C

$$u(0, y) = u(y, 0) = 0,\ u(1.5, y) = 3y^3$$

Neumann B.C

$$u_y(x, 1) = 6x$$

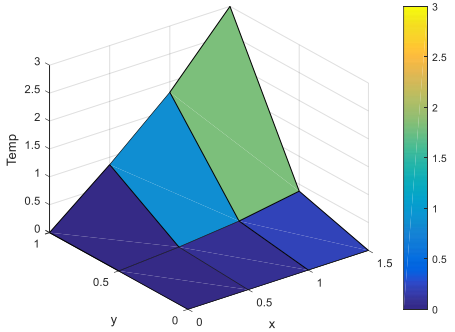
1) $h = 0.5$

```
u12 =          u22 =

    0.8665         1.8121

u11 =          u21 =

    0.0769         0.1910
```



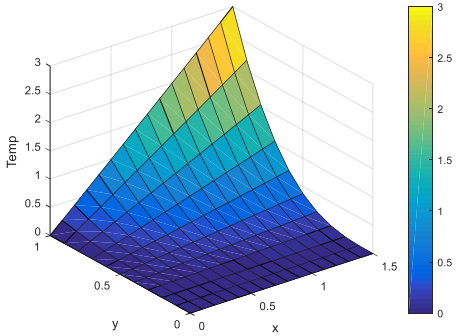
2) $h = 0.1$

```
u12 =          u22 =

    0.9941         1.9910

u11 =          u21 =

    0.1229         0.2474
```

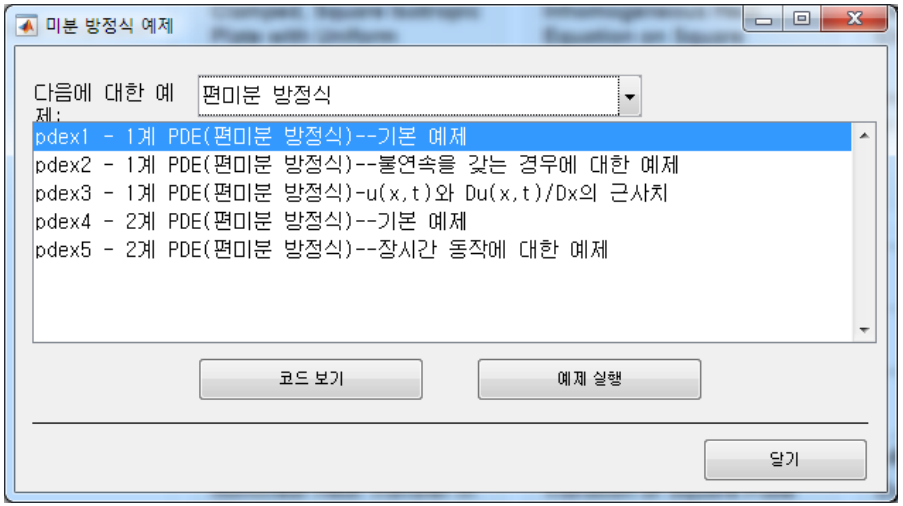


Ref. : Advanced Engineering Mathematics, 9th edition, Chap. 21, pp918-919

ODEEXAMPLES

명령 창

```
>> odeexamples
```




```
sol = pdepe(m,@pdex1pde,@pdex1ic,@pdex1bc,x,t);
```

pdepe

Solve initial-boundary value problems for parabolic-elliptic PDEs in 1-D

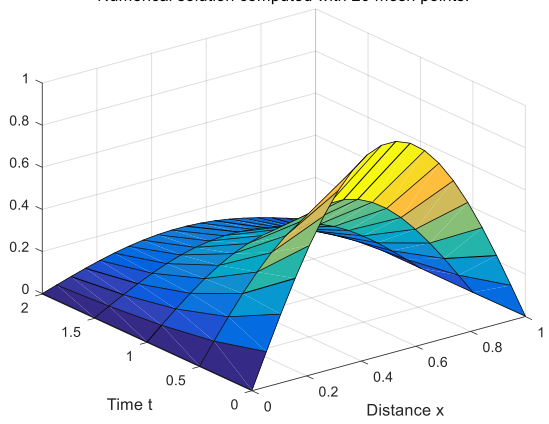
pdepe solves PDEs of the form:

$$c\left(x,t,u,\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial t}=x^{-m}\frac{\partial}{\partial x}\left(x^{m}f\left(x,t,u,\frac{\partial u}{\partial x}\right)\right)+s\left(x,t,u,\frac{\partial u}{\partial x}\right)$$

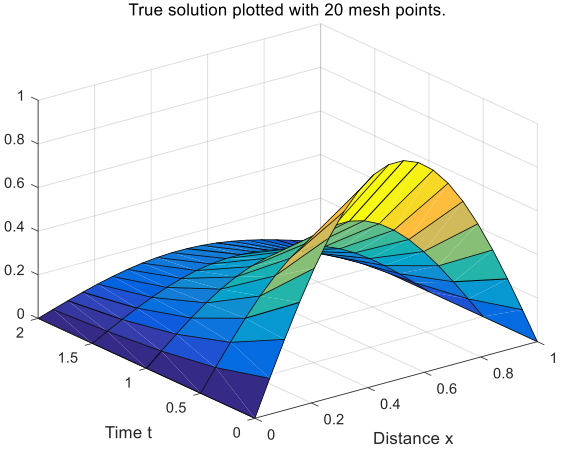
```
function [c,f,s] = pdex1pde(x,t,u,DuDx)
c = pi^2;
f = DuDx;
s = 0;
```

$$\pi^2\frac{\partial u}{\partial t}=\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right)$$

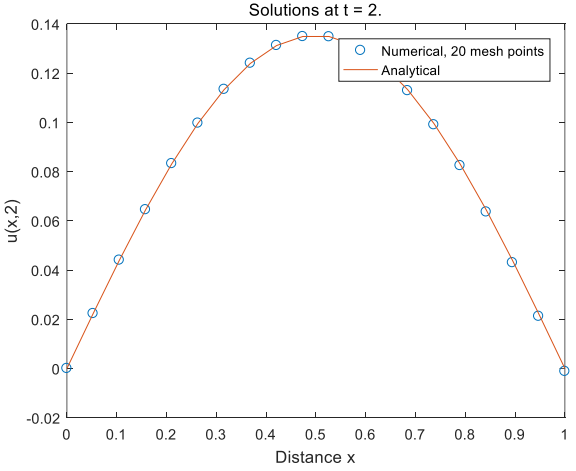
Numerical solution computed with 20 mesh points.



True solution plotted with 20 mesh points.



Solutions at t = 2.



PDE TOOLBOX IN MATLAB

MATLAB Examples

Search Examples | Examples

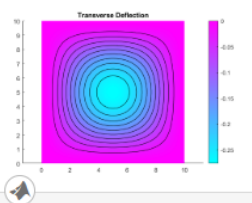
CATEGORY | Close

- Examples Home
- MATLAB Family
- Math, Statistics, and Optimization

Partial Differential Equation Toolbox

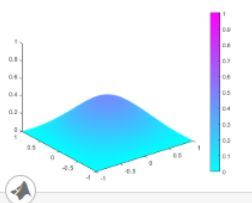
- Getting Started with Partial Differential Equation Toolbox
- Applications
- PDE Problem Setup
- Solution Visualization and Interpolation
- Other

Applications



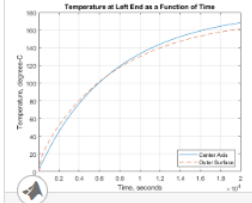
Clamped, Square Isotropic Plate with Uniform Pressure Load

Calculate the deflection of a structural plate acted on by a pressure loading.



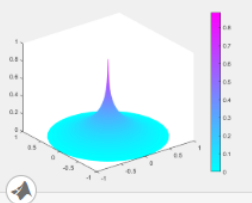
Inhomogeneous Heat Equation on Square Domain

Solve the heat equation with a source term.



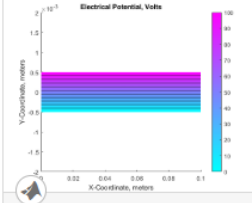
Heat Distribution in Circular Cylindrical Rod

Analyze a 3-D axisymmetric model by using a 2-D model.



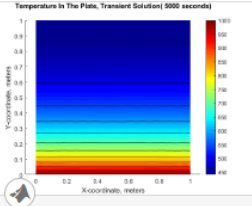
Poisson's Equation with Point Source and Adaptive Mesh Refinement

Solve a Poisson's equation with a delta-function point source on the unit disk using the adaptmesh function.



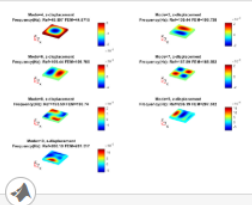
Deflection of Piezoelectric Actuator

Solve a coupled elasticity-electrostatics problem.



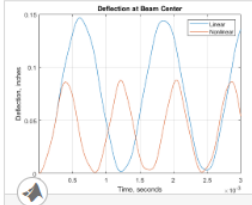
Nonlinear Heat Transfer in Thin Plate

Perform a heat transfer analysis of a thin plate.



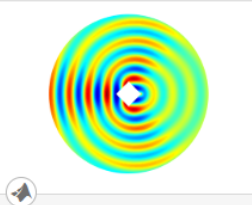
Vibration of Square Plate

Calculate the vibration modes and frequencies of a 3-D simply supported, square, elastic plate.



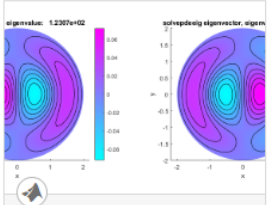
Dynamic Analysis of Clamped Beam

Analyze the dynamic behavior of a beam clamped at both ends and loaded with a uniform pressure load.



Helmholtz's Equation on Unit Disk with Square Hole

Solve a Helmholtz equation using the solvepde function.



Vibration of Circular Membrane

Calculate the vibration modes of a circular membrane by using the MATLAB eigs function.

APPENDIX

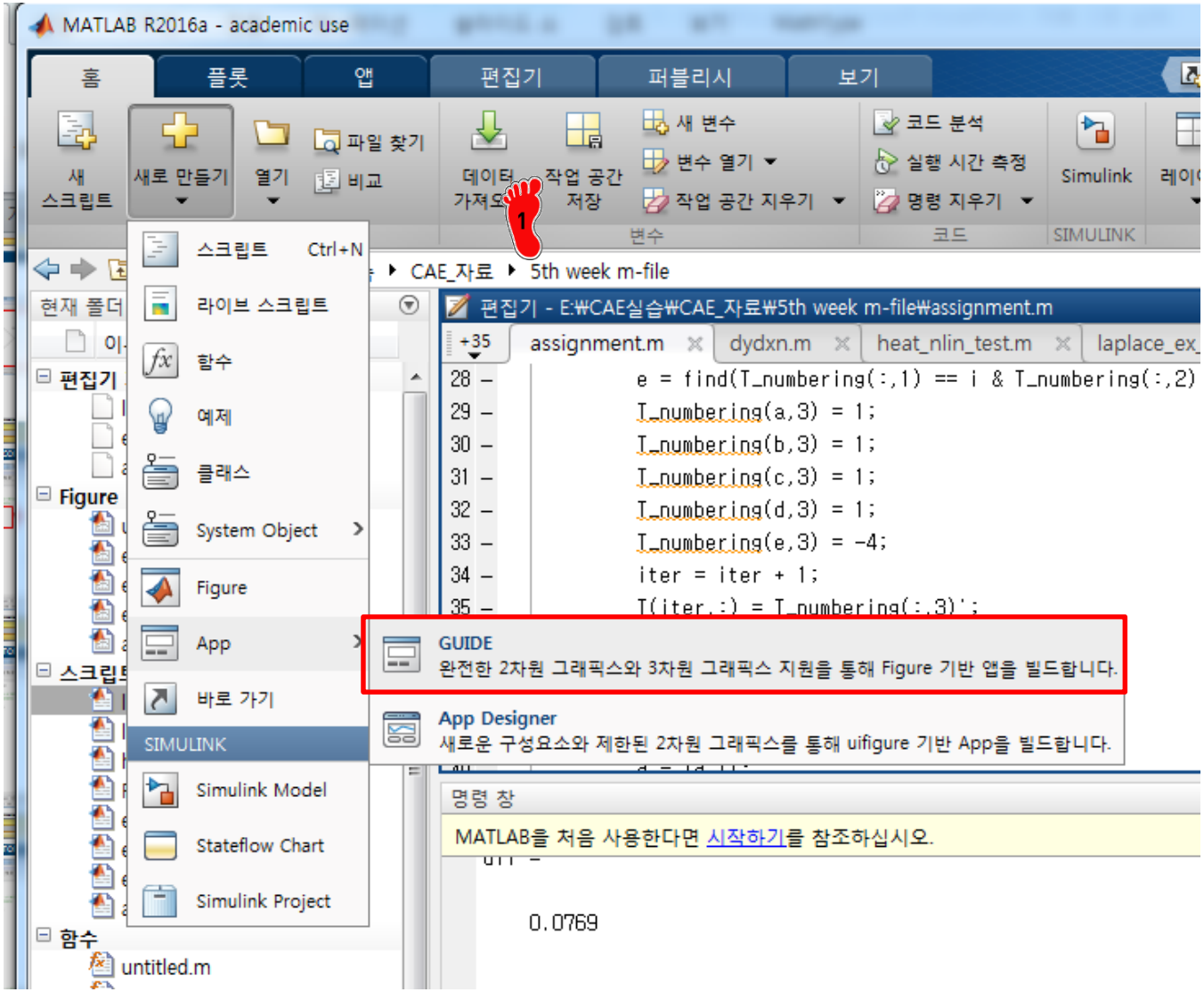
- **MATLAB GUI implementation**

  - ✓ **GUI**

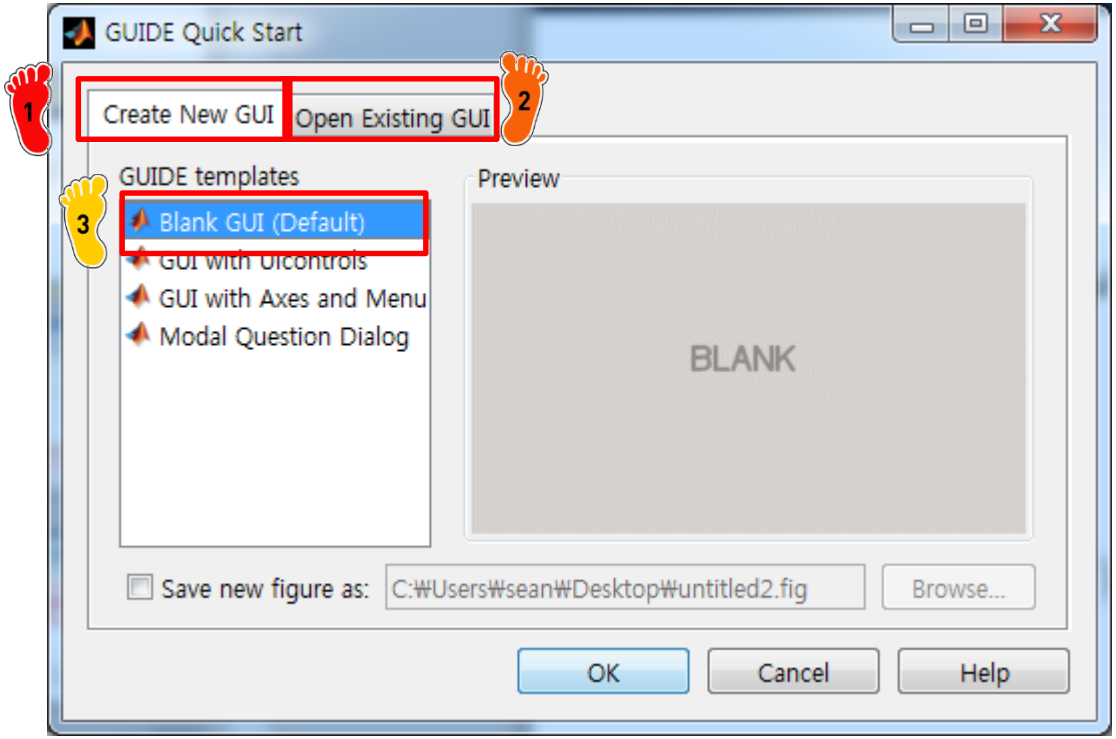
  - ✓ **Example**

START

1 home – New – APP - GUIDE 클릭



GUI SETTING WINDOW



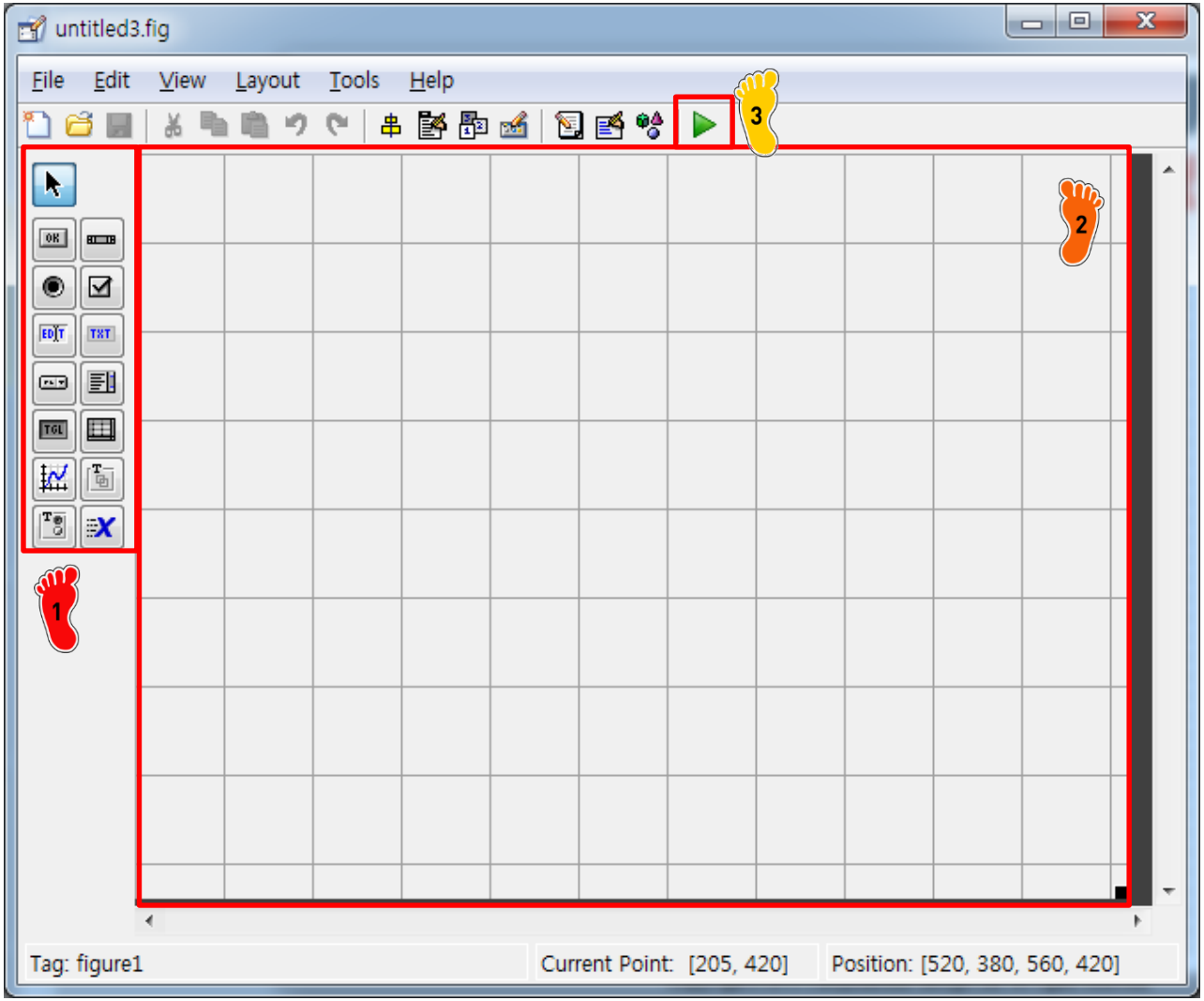
1 새로운 GUI 를 만드는 탭 메뉴

2 기존 GUI 파일을 불러오는 탭 메뉴

3 빈 GUI 창 생성(기본)

그 아래 메뉴는 예제

GUI WINDOW

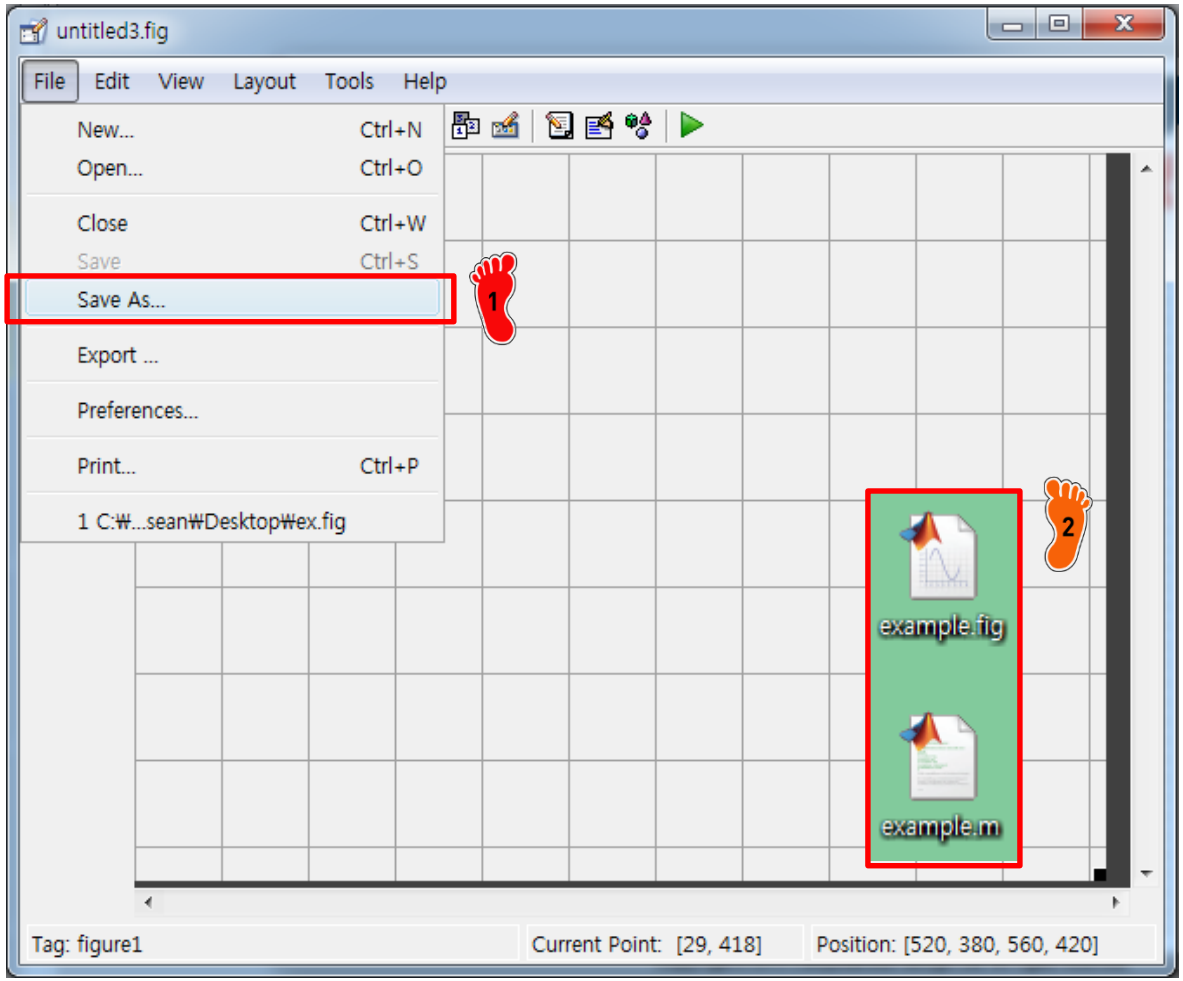


1 GUI 아이콘

2 GUI 아이콘을 배치하는 창

3 GUI 실행 아이콘

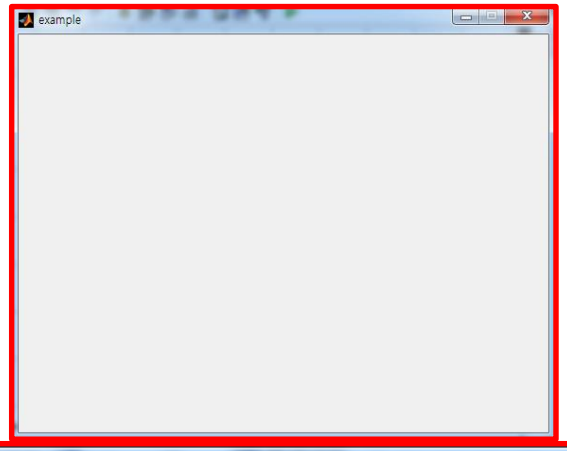
GUI FILES

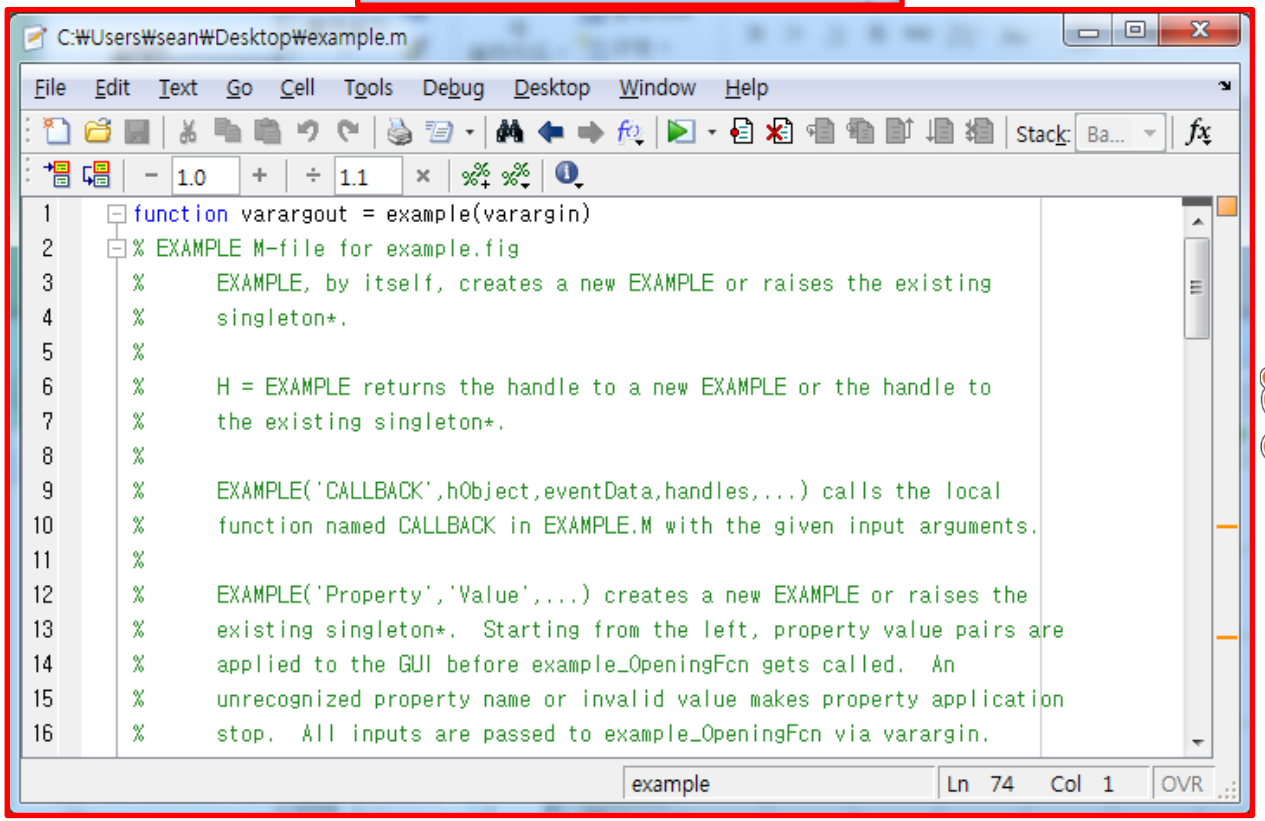


1 현재 GUI 창을 저장

2 저장하면 *.fig 파일과 *.m 파일이 동시에 생성됨

GUI M-FILE





1. GUI 를 실행하면 설정한 아이콘이 없기 때문에 빈 창이 pop up 됨

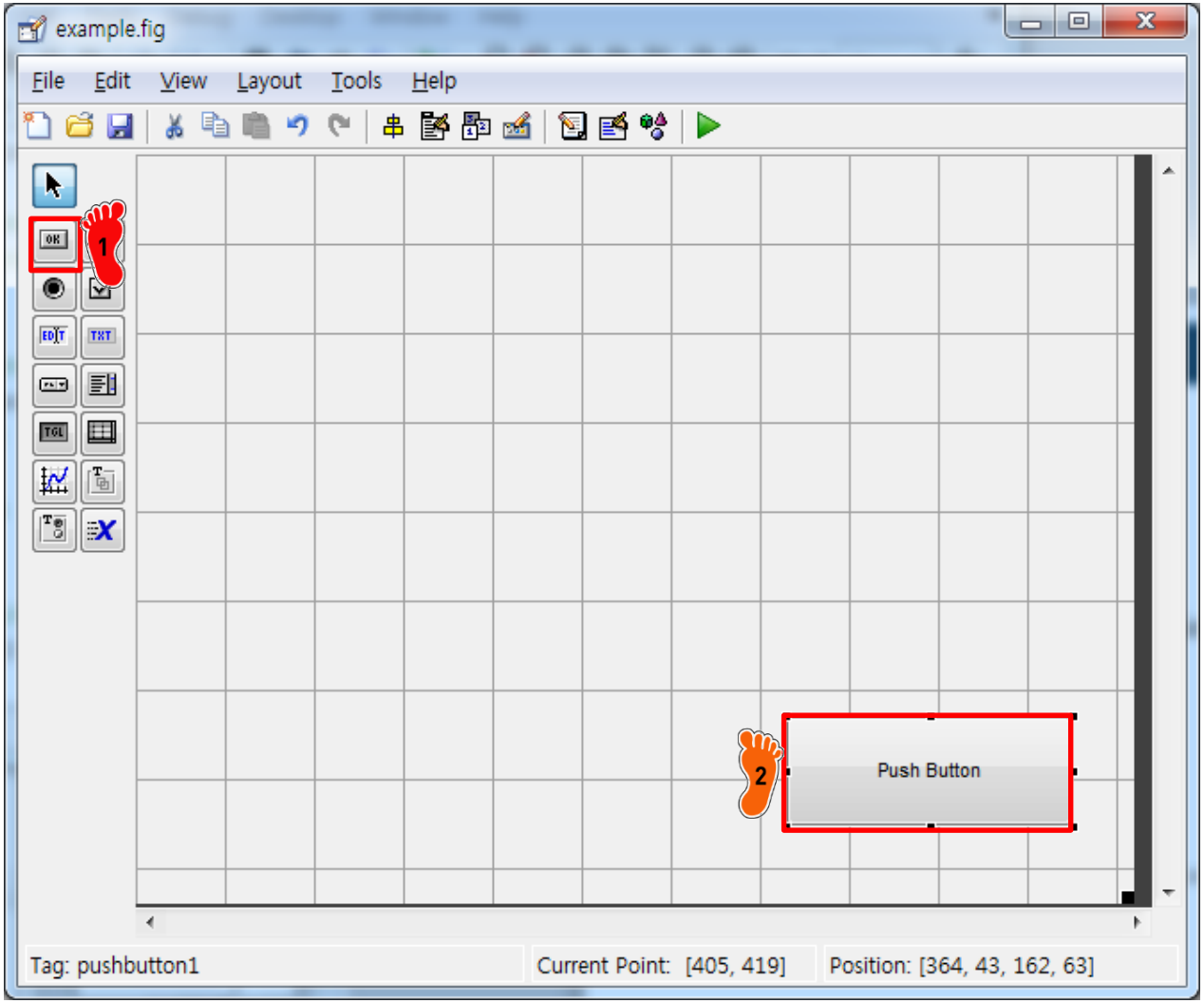
2. m-file 내용은 GUI 를 이용하는 방법 설명과 현재 GUI 창을 pop up 시키는 명령어로 이루어짐

APPENDIX

- **MATLAB GUI implementation**

  - ✓ **GUI**
  - ✓ **Example (Prob.27-27에 참고)**
    - ✓ **ODE 문제를 입력하여 다양한 방법으로 수치해를 구하고 비교할 수 있는 GUI 구축**

PUSH BUTTON

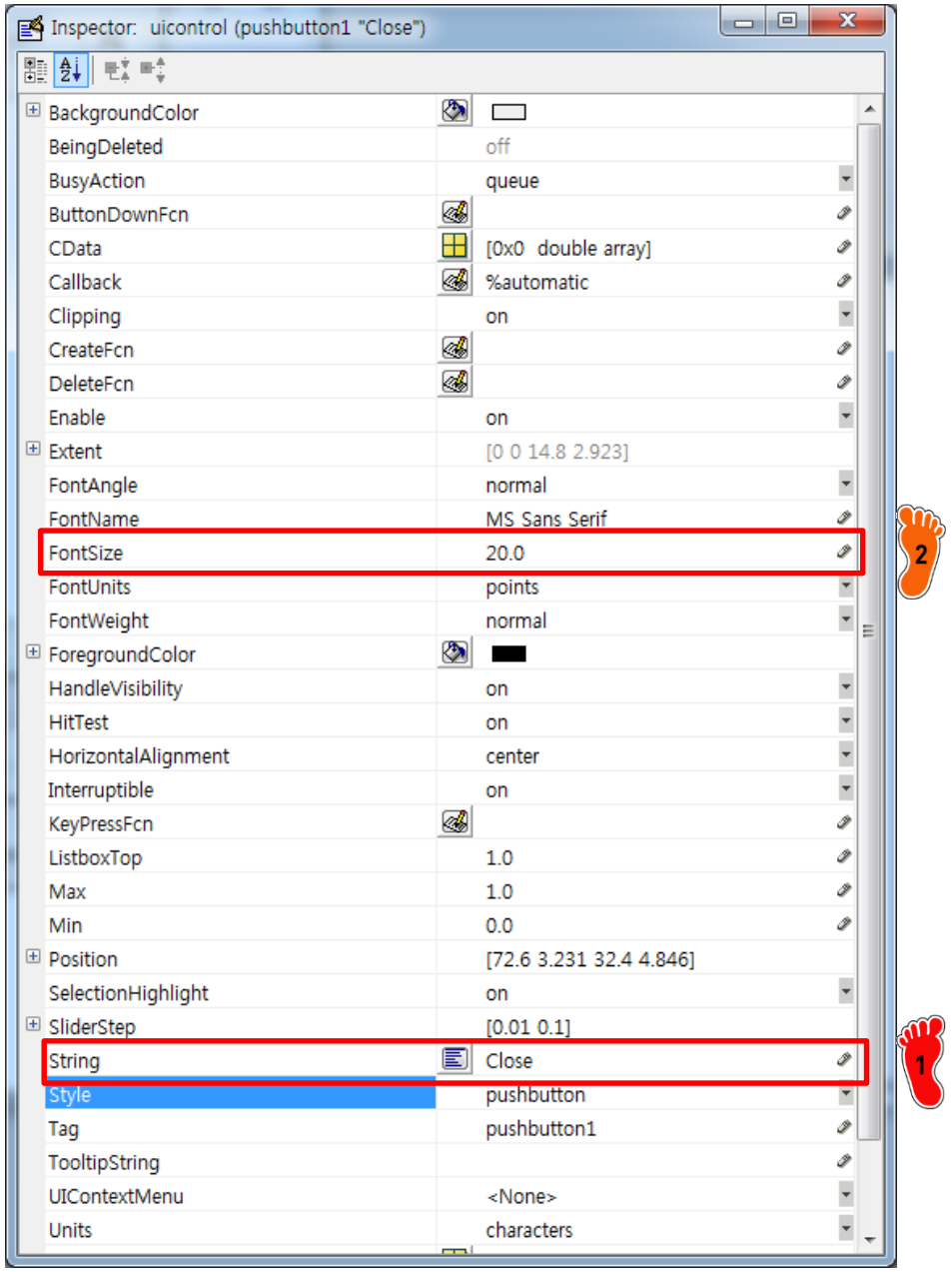


1 Push Button 아이콘 클릭

2 마우스를 드래그해서 Push Button 아이콘 생성

그 후 아이콘 더블 클릭

PUSH BUTTON: INSPECTOR



1 String 을 Close 로 변경

2 FontSize 20 으로 변경

PUSH BUTTON: M-FILE

C:₩Users₩sean₩Desktop₩example.m

File Edit Text Go Cell Tools Debug Desktop Window Help

```
60
61    % UIWAIT makes example wait for user response (see UIRESUME)
62    % uiwait(handles.figure1);
63
64
65    % --- Outputs from this function are returned to the command line.
66    function varargout = example_OutputFcn(hObject, eventdata, handles)
67    % varargout  cell array for returning output args (see VARARGOUT);
68    % hObject    handle to figure
69    % eventdata  reserved - to be defined in a future version of MATLAB
70    % handles    structure with handles and user data (see GUIDATA)
71
72    % Get default command line output from handles structure
73    varargout{1} = handles.output;
74
75
76    % --- Executes on button press in pushbutton1.
77    function pushbutton1_Callback(hObject, eventdata, handles)
78    % hObject    handle to pushbutton1 (see GCBO)
79    % eventdata  reserved - to be defined in a future version of MATLAB
80    % handles    structure with handles and user data (see GUIDATA)
81
```

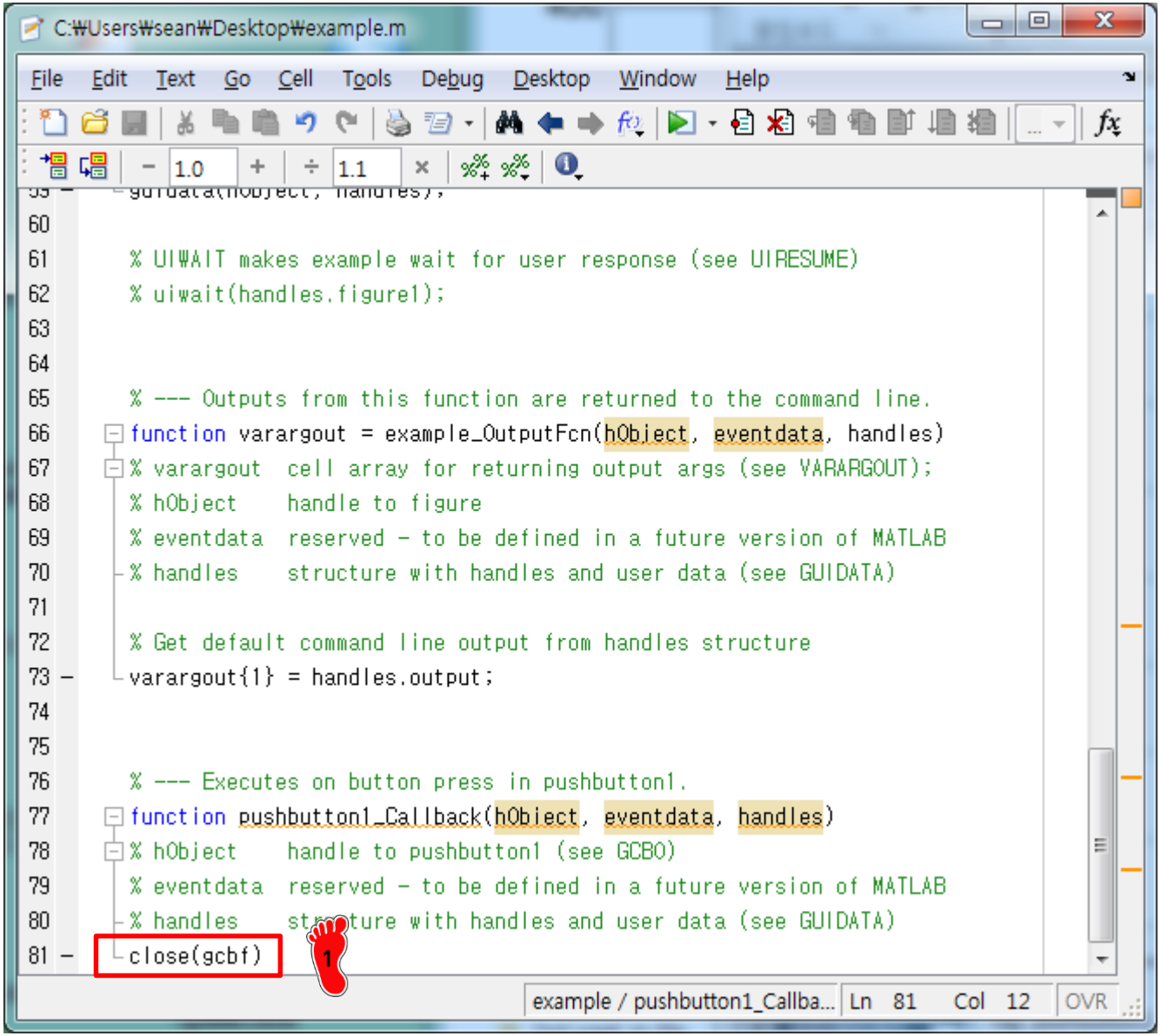
example / example_OutputFcn Ln 73 Col 31 OVR

1

1 M-file 에 pushbutton1_Callback 함수가 생성됨

이 함수 밑에 m-file 명령어를 입력하고 저장하면 pushbutton 을 클릭했을 때 실행됨

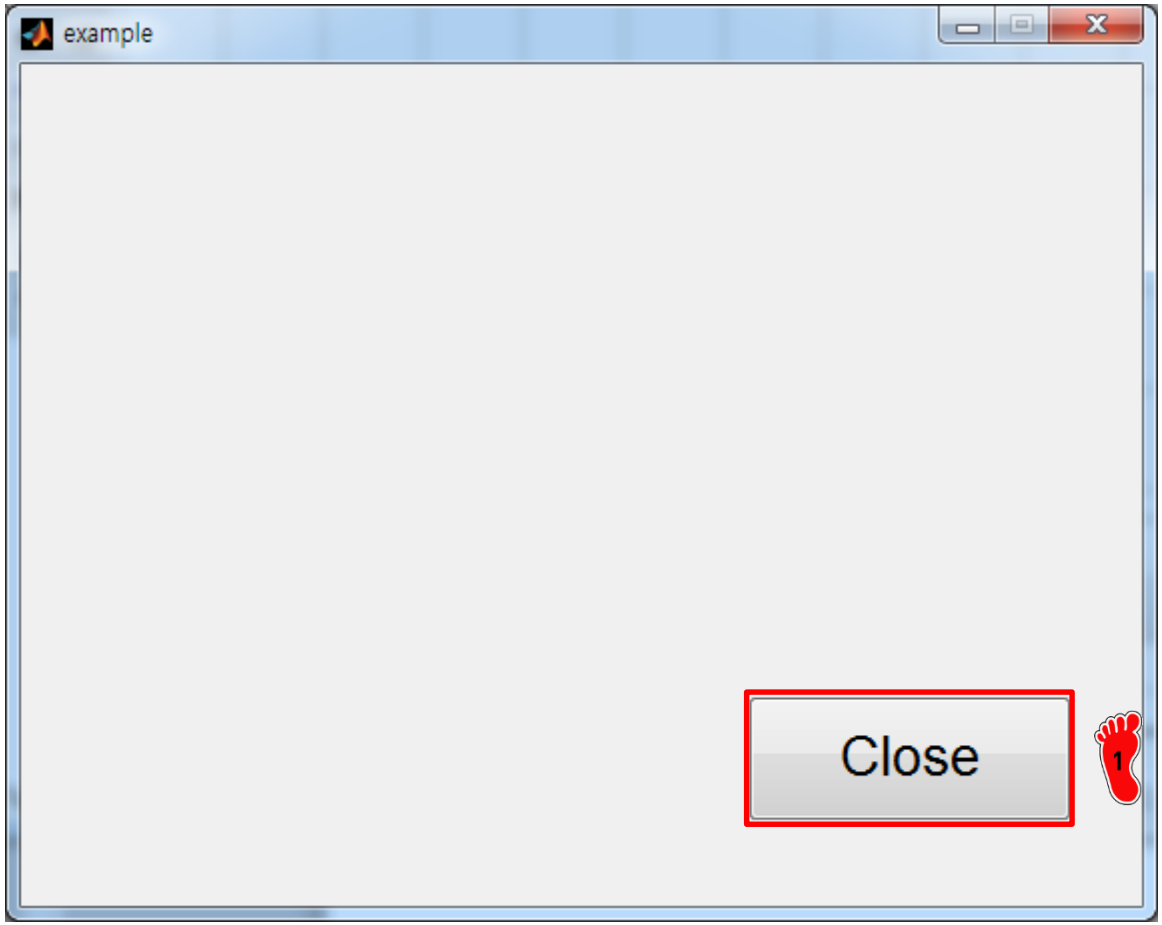
PUSH BUTTON: M-FILE



1 close(gcbf) 입력

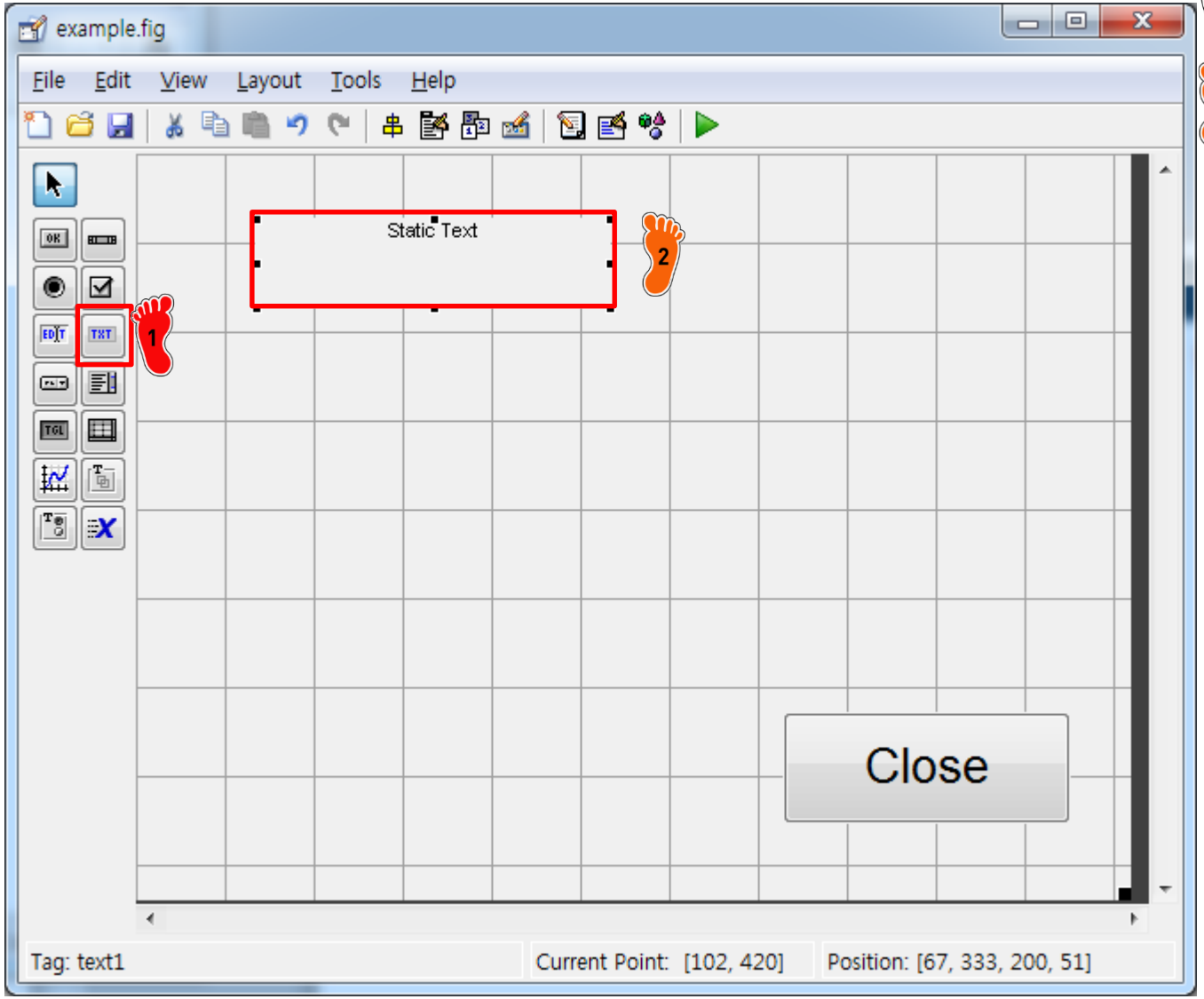
이 명령어는 현재 GUI 창을 닫는 명령어

PUSH BUTTON: GUI



1 GUI 창을 실행시켜 Close pushbutton 을 클릭하면 창이 닫힘

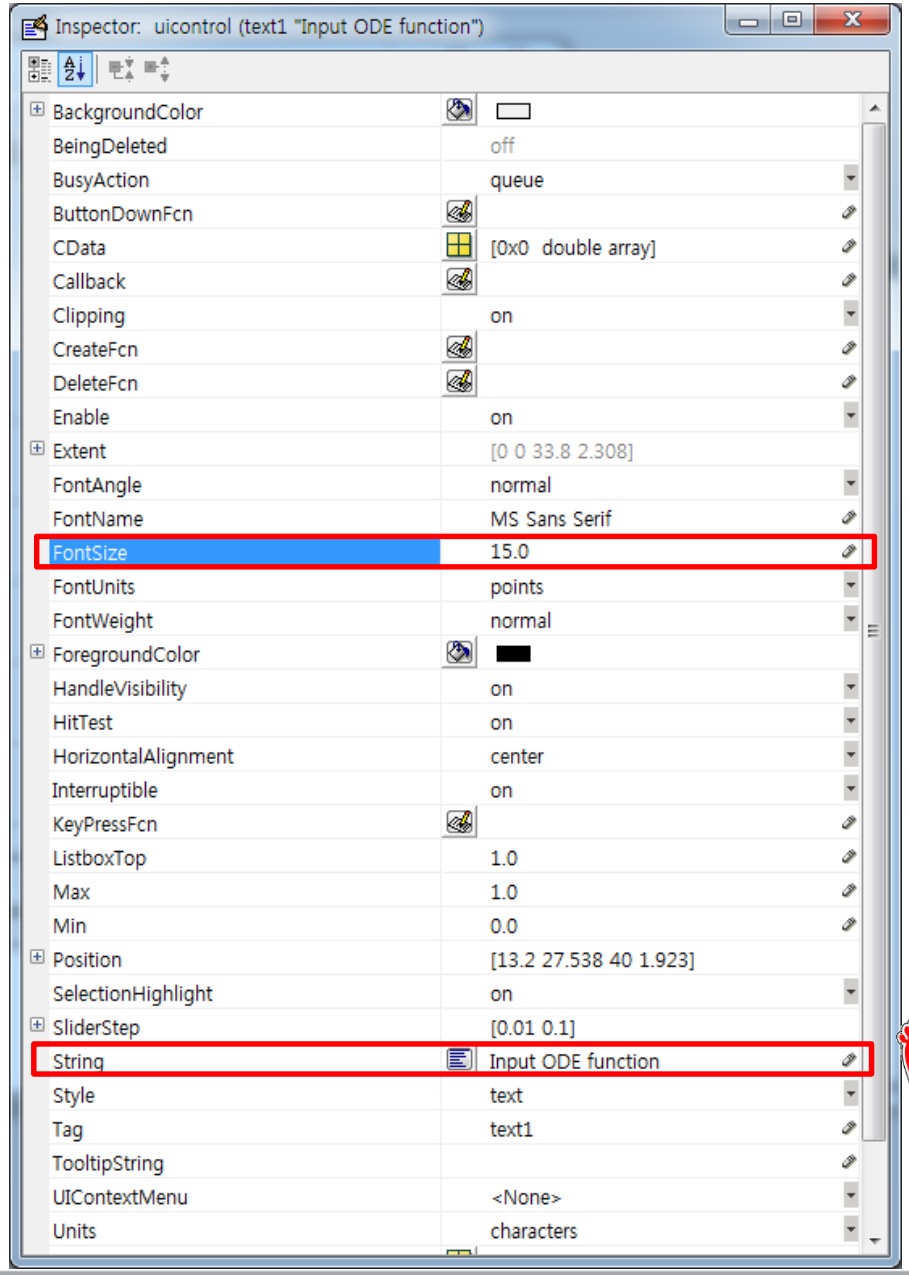
STATIC TEXT BOX



1 static text 아이콘 클릭

2 static text box 생성
더블 클릭

STATIC TEXT BOX: INSPECTOR

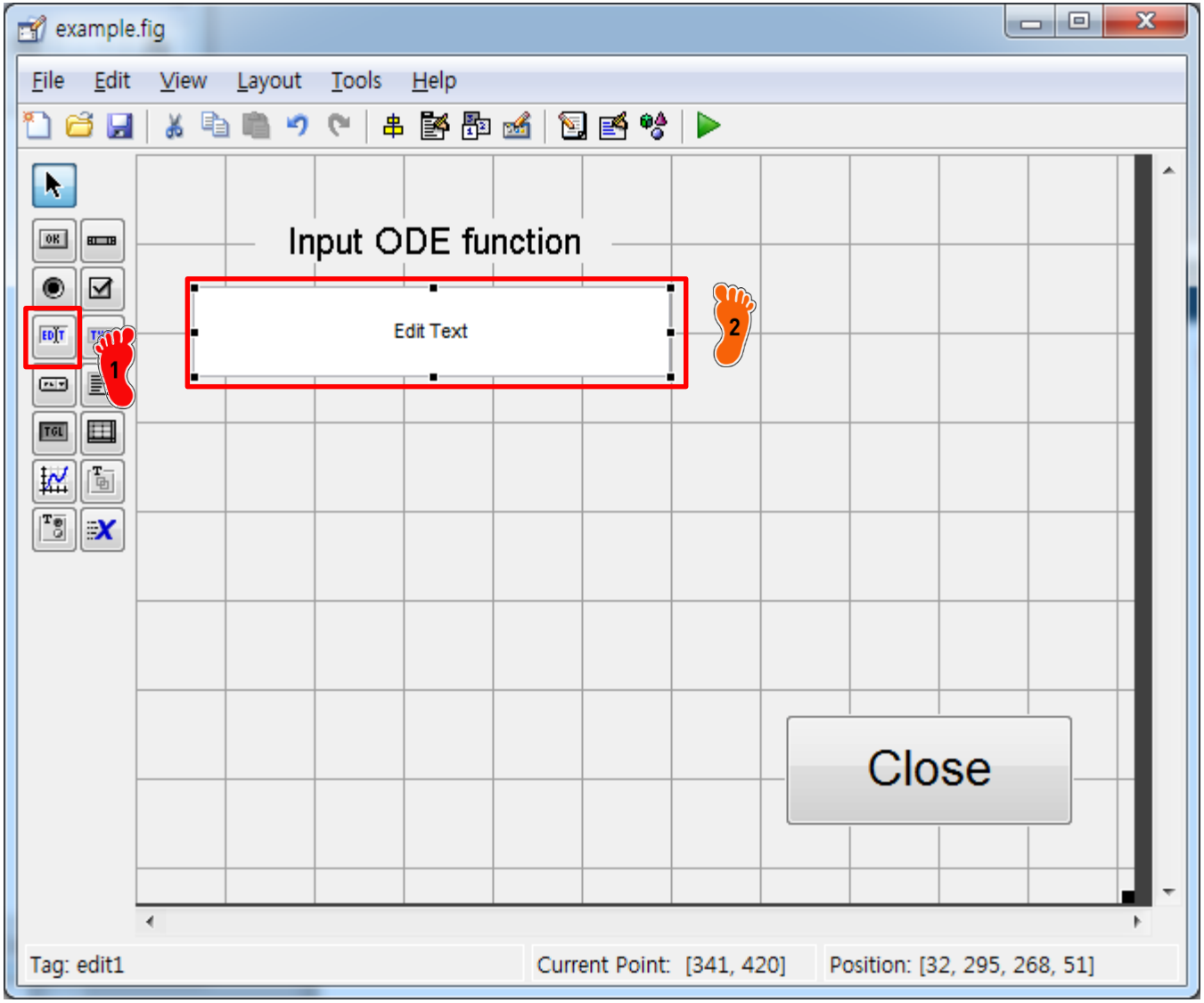


1

1 String 을 Input ODE function 으로 변경

2 FontSize 15로 변경

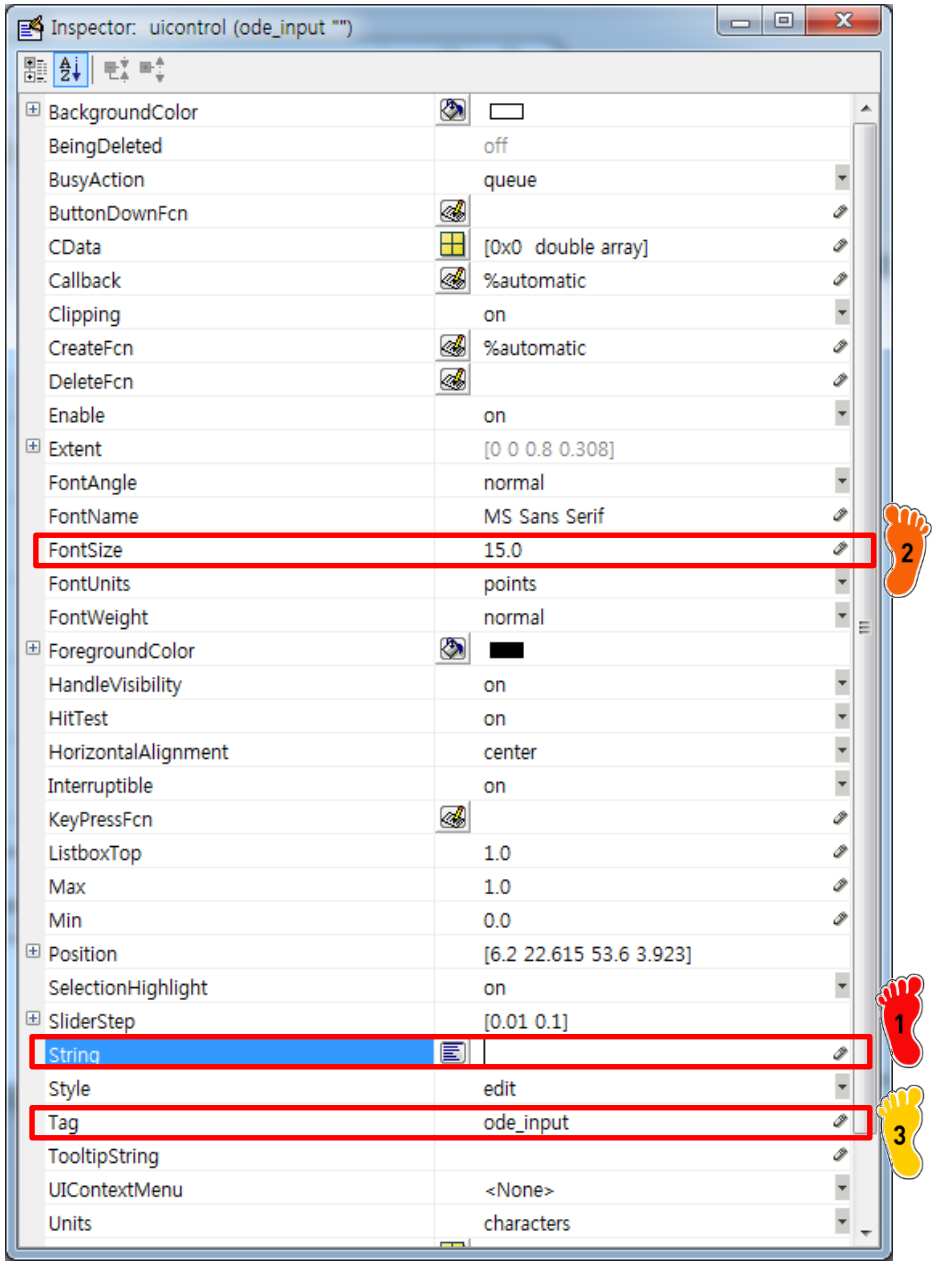
EDIT TEXT BOX



1 edit text 아이콘 클릭

2 edit text box 생성
더블 클릭

EDIT TEXT BOX: INSPECTOR

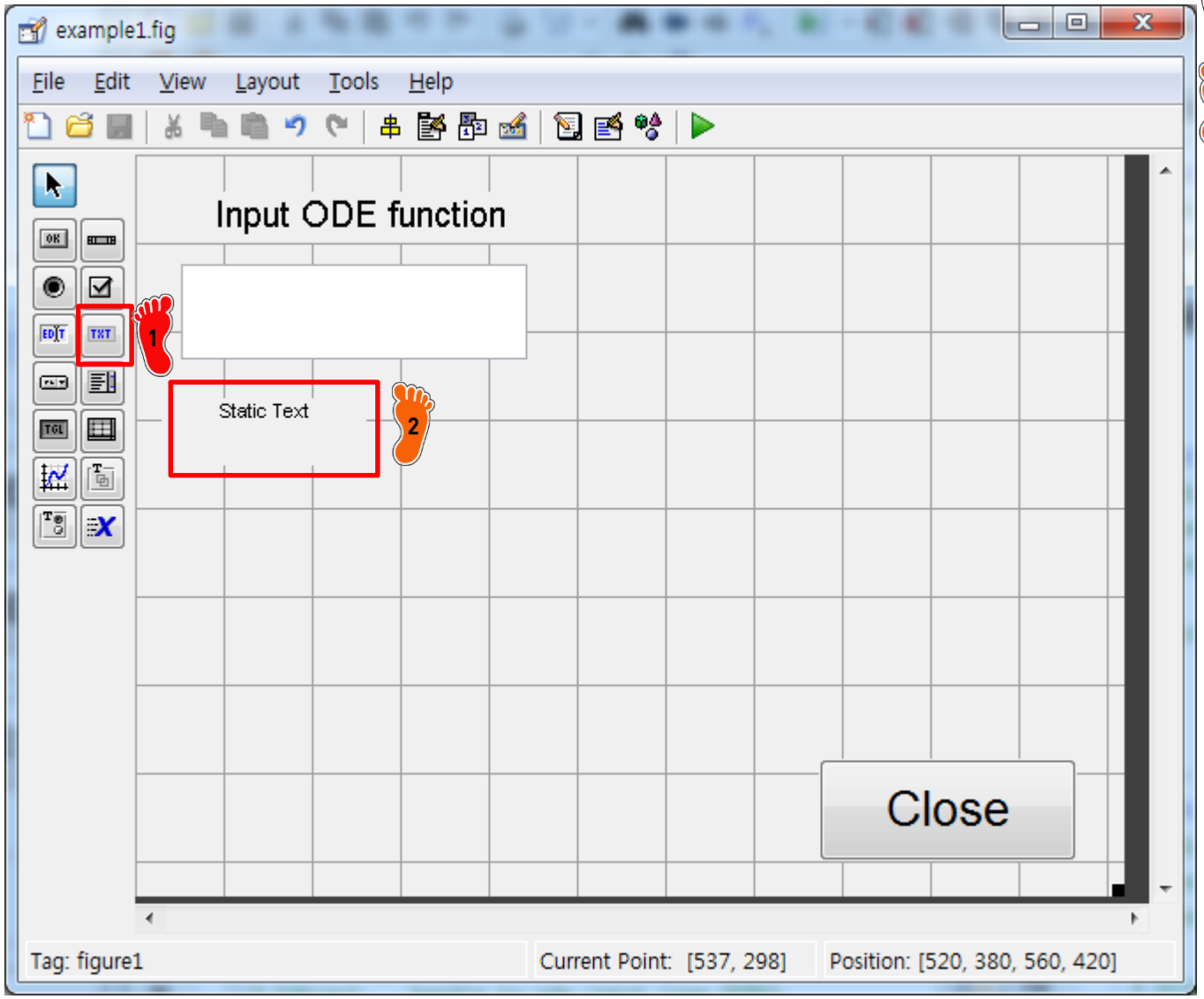


1 String 빈칸으로 변경

2 FontSize 15로 변경

3 Tag 를 ode_input 으로 변경

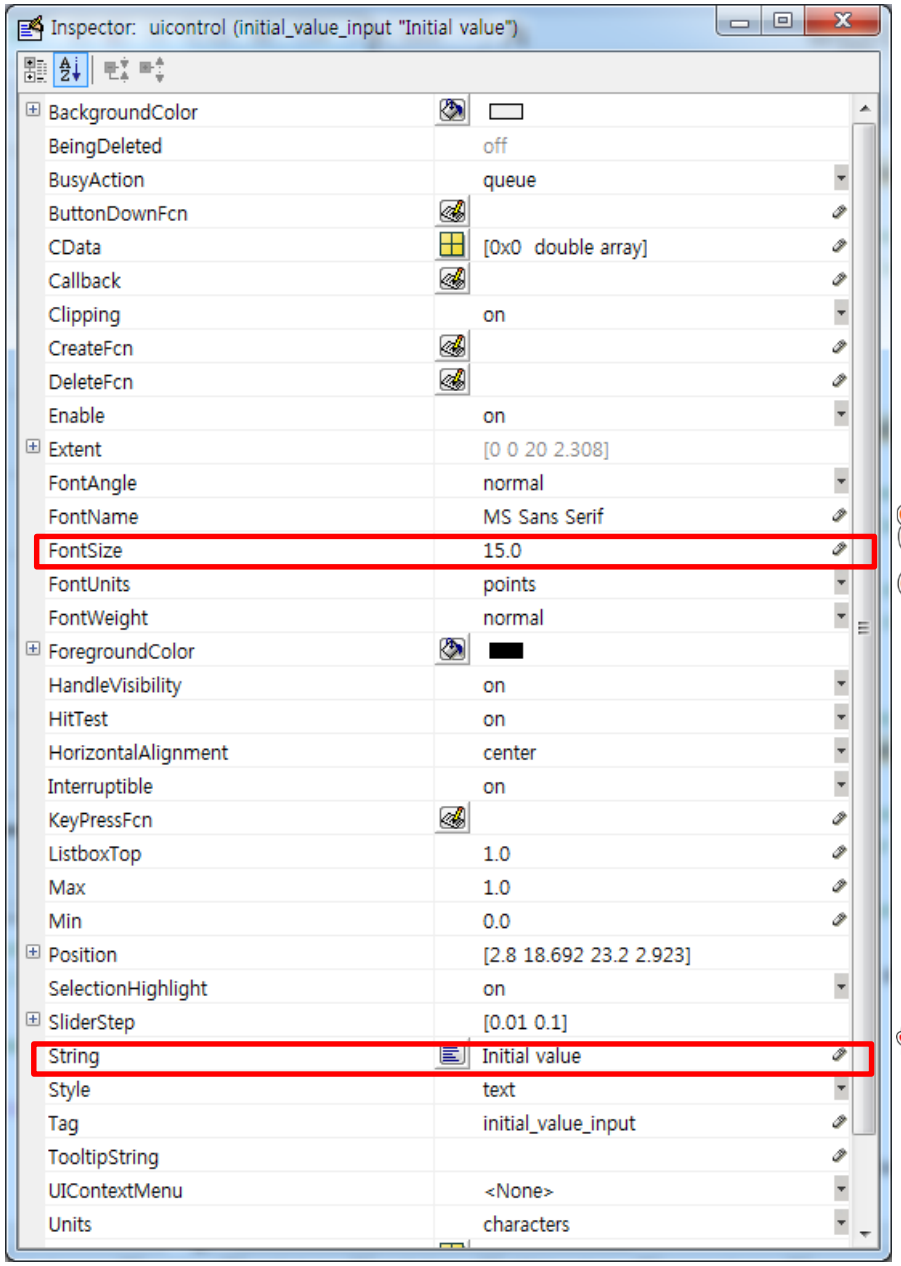
STATIC TEXT BOX



1 static text 아이콘 클릭

2 static text box 생성
더블 클릭

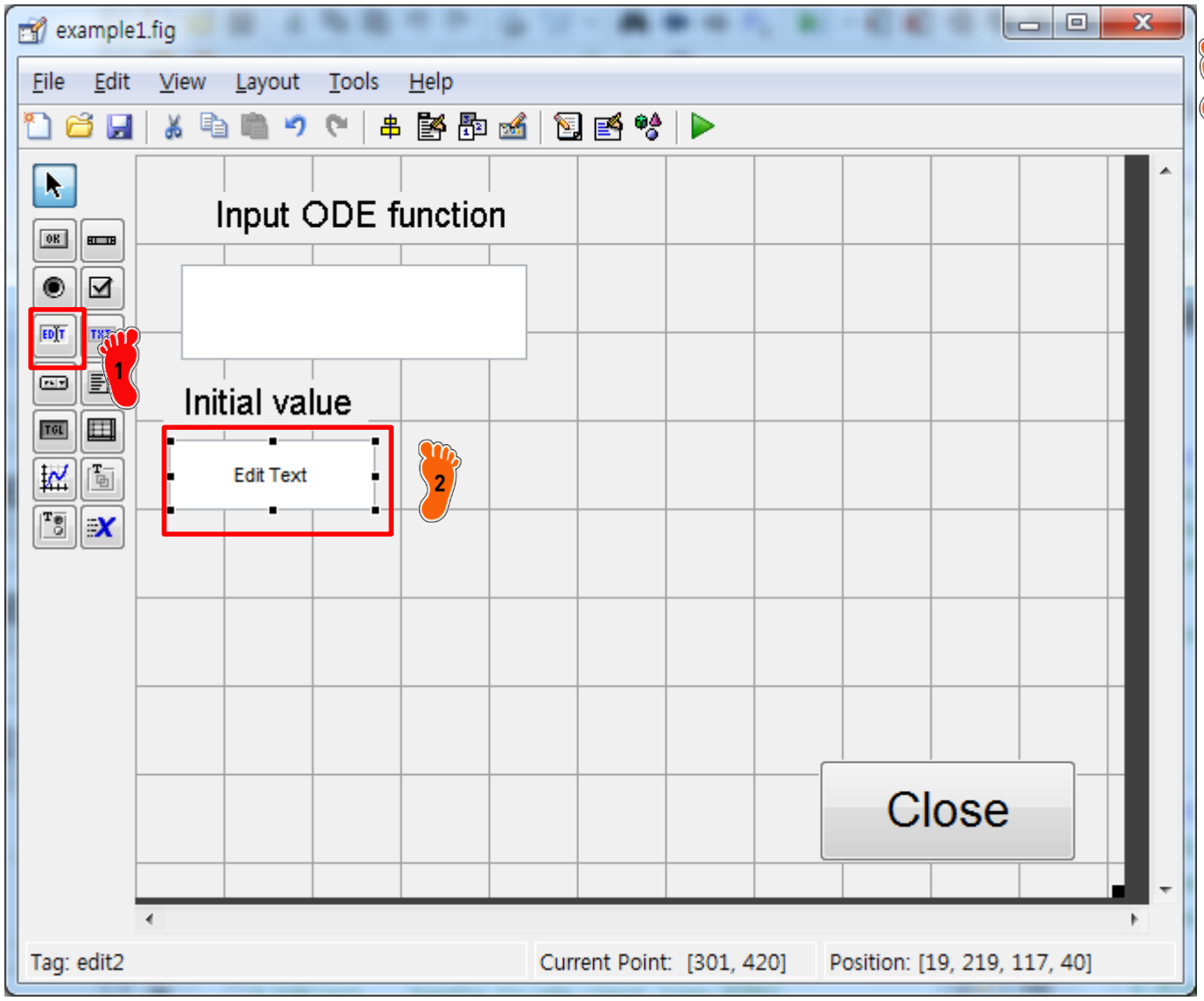
STATIC TEXT BOX: INSPECTOR



1 String 을 Initial value 로 변경

2 FontSize 15로 변경

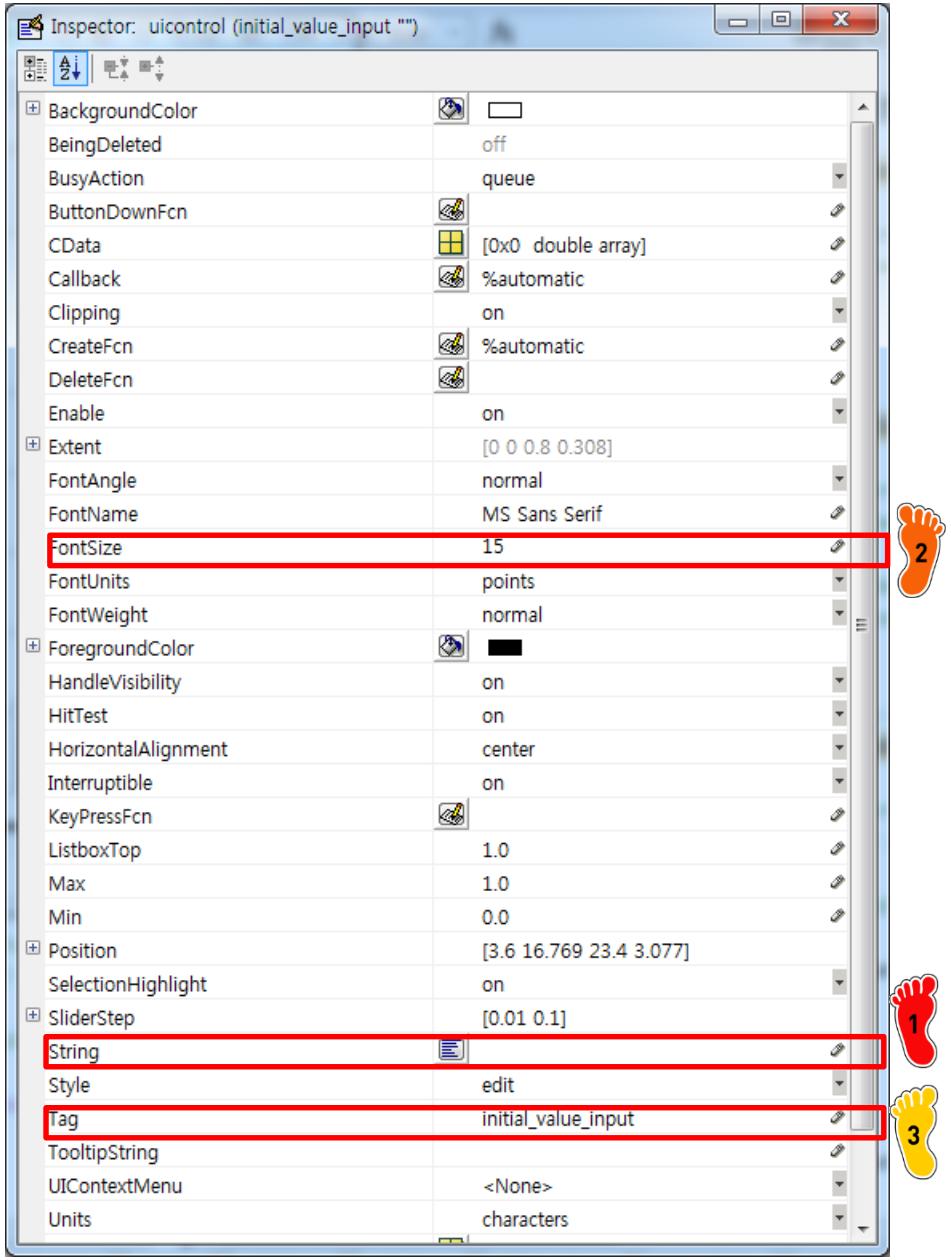
EDIT TEXT BOX



1 edit text 아이콘 클릭

2 edit text box 생성
더블 클릭

EDIT TEXT BOX: INSPECTOR

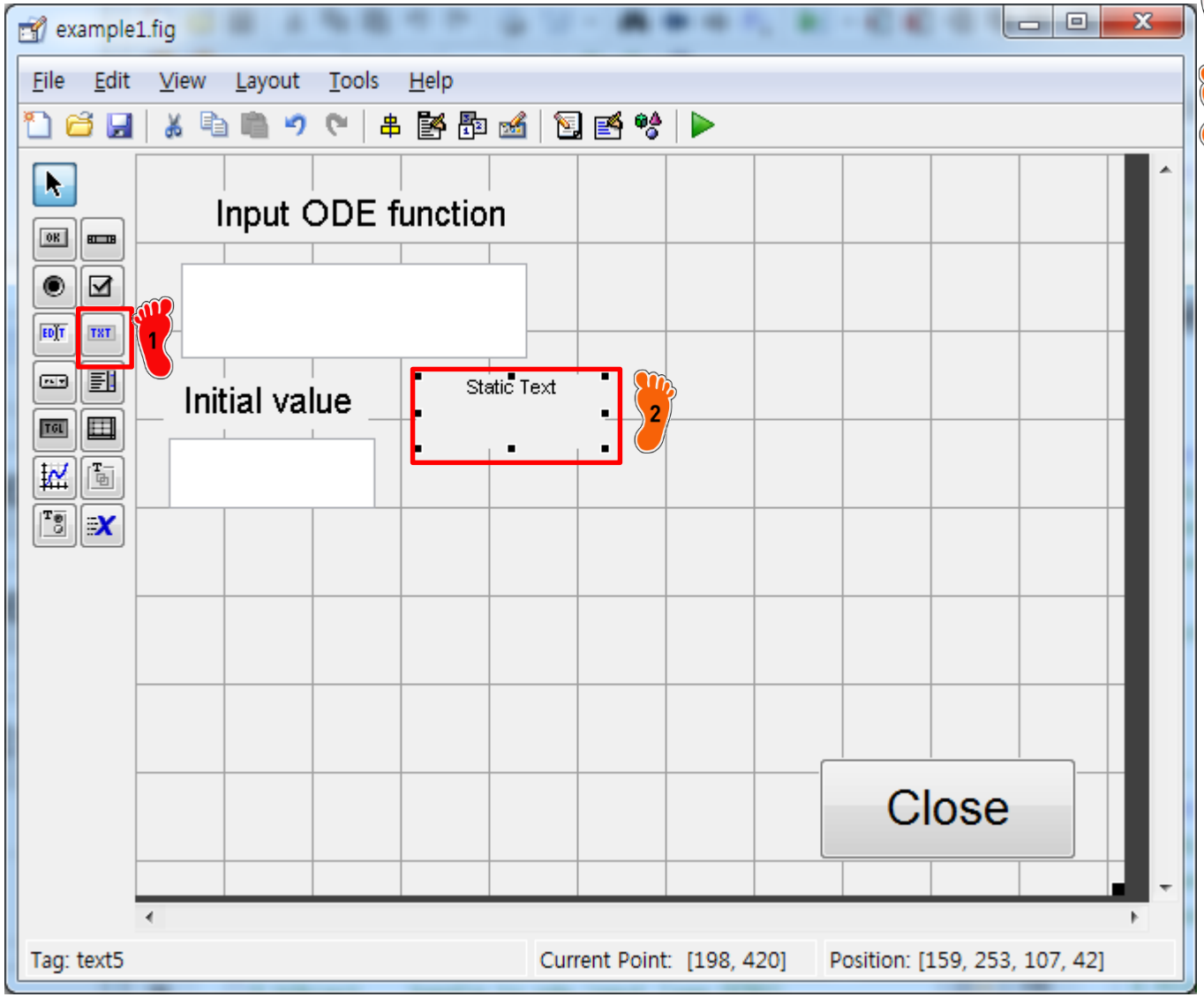


1 String 빈칸으로 변경

2 FontSize 15로 변경

3 Tag 를 initial_value_input 으로 변경

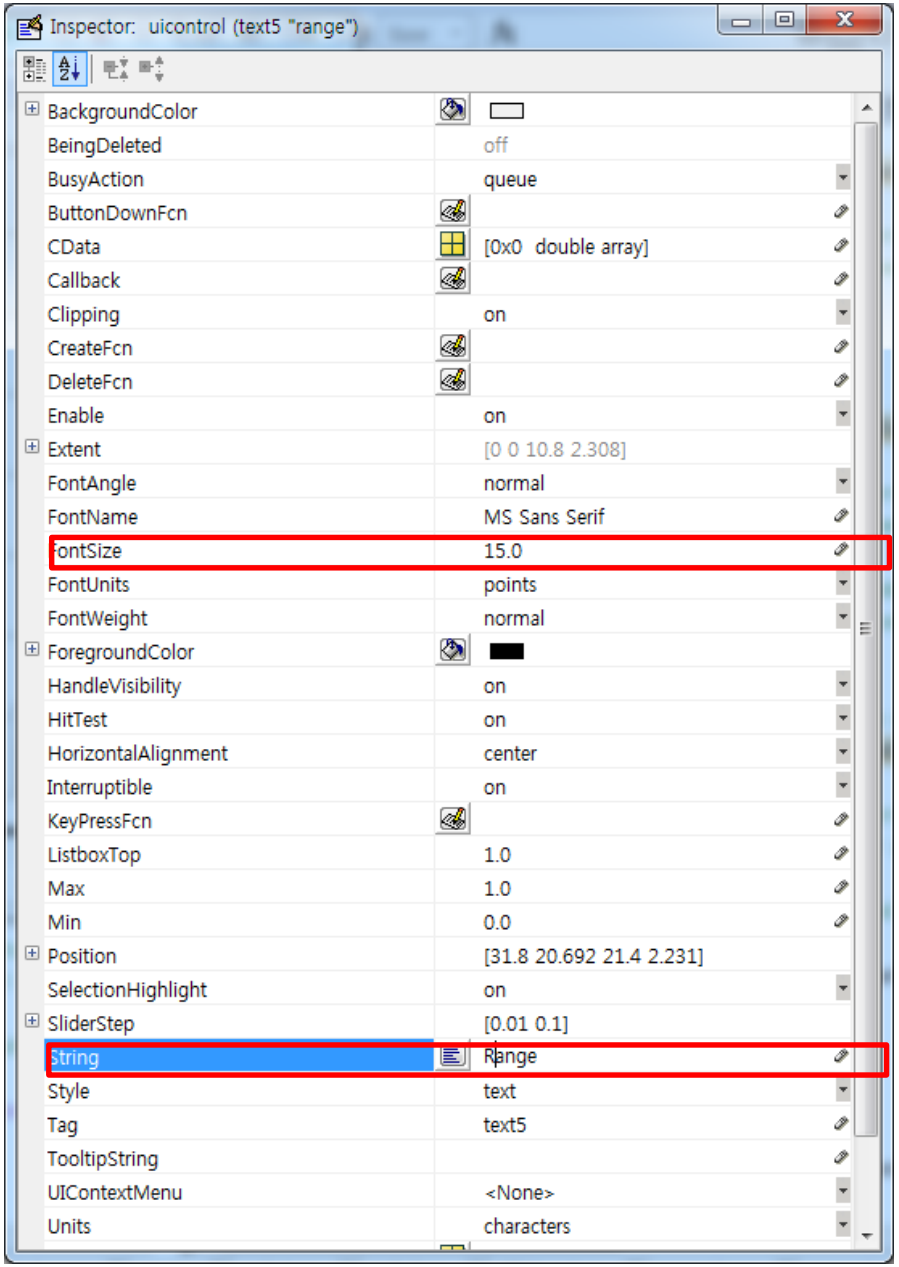
STATIC TEXT BOX



1 static text 아이콘 클릭

2 static text box 생성
더블 클릭

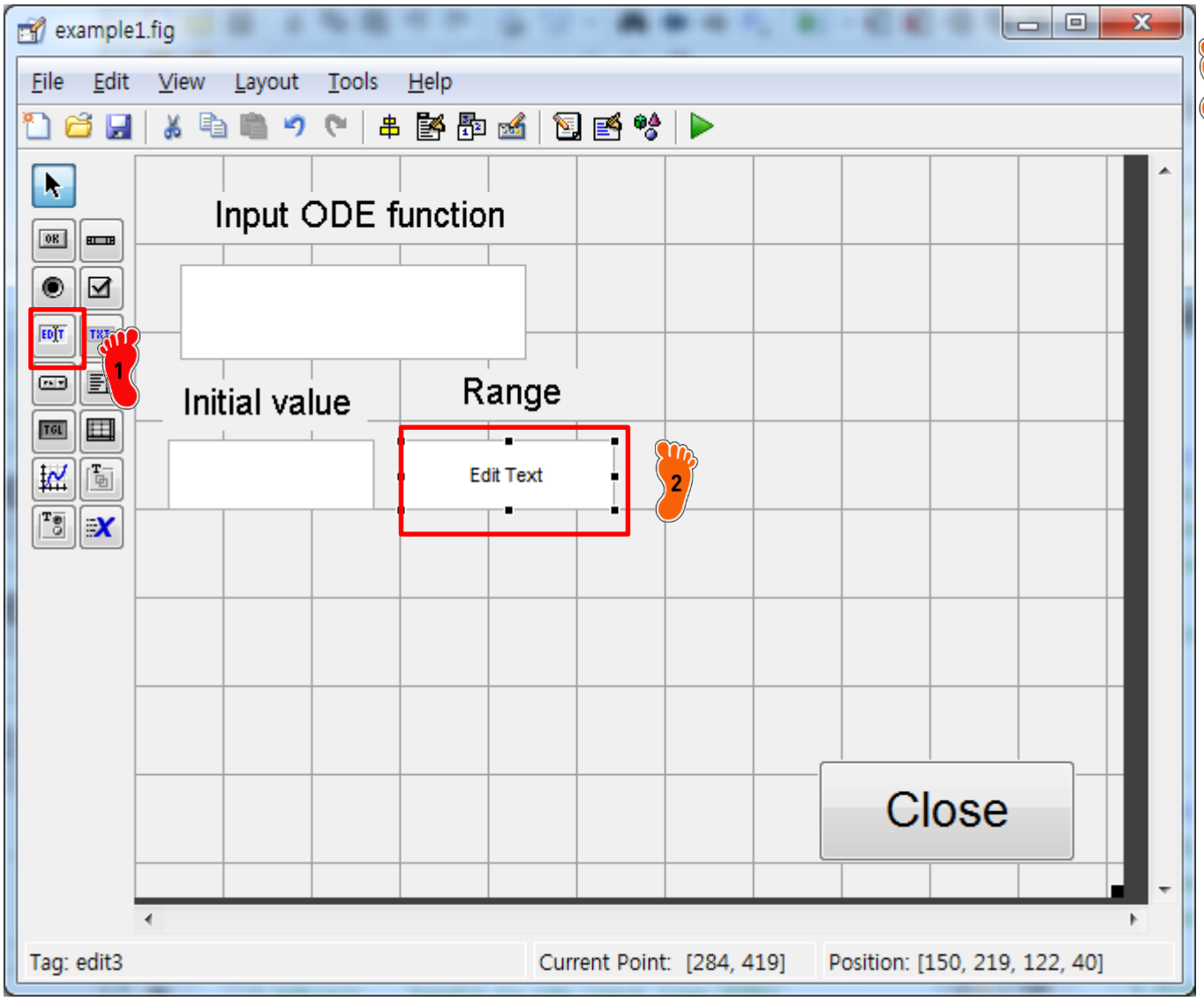
STATIC TEXT BOX: INSPECTOR



1 String 을 Range 로 변경

2 FontSize 15로 변경

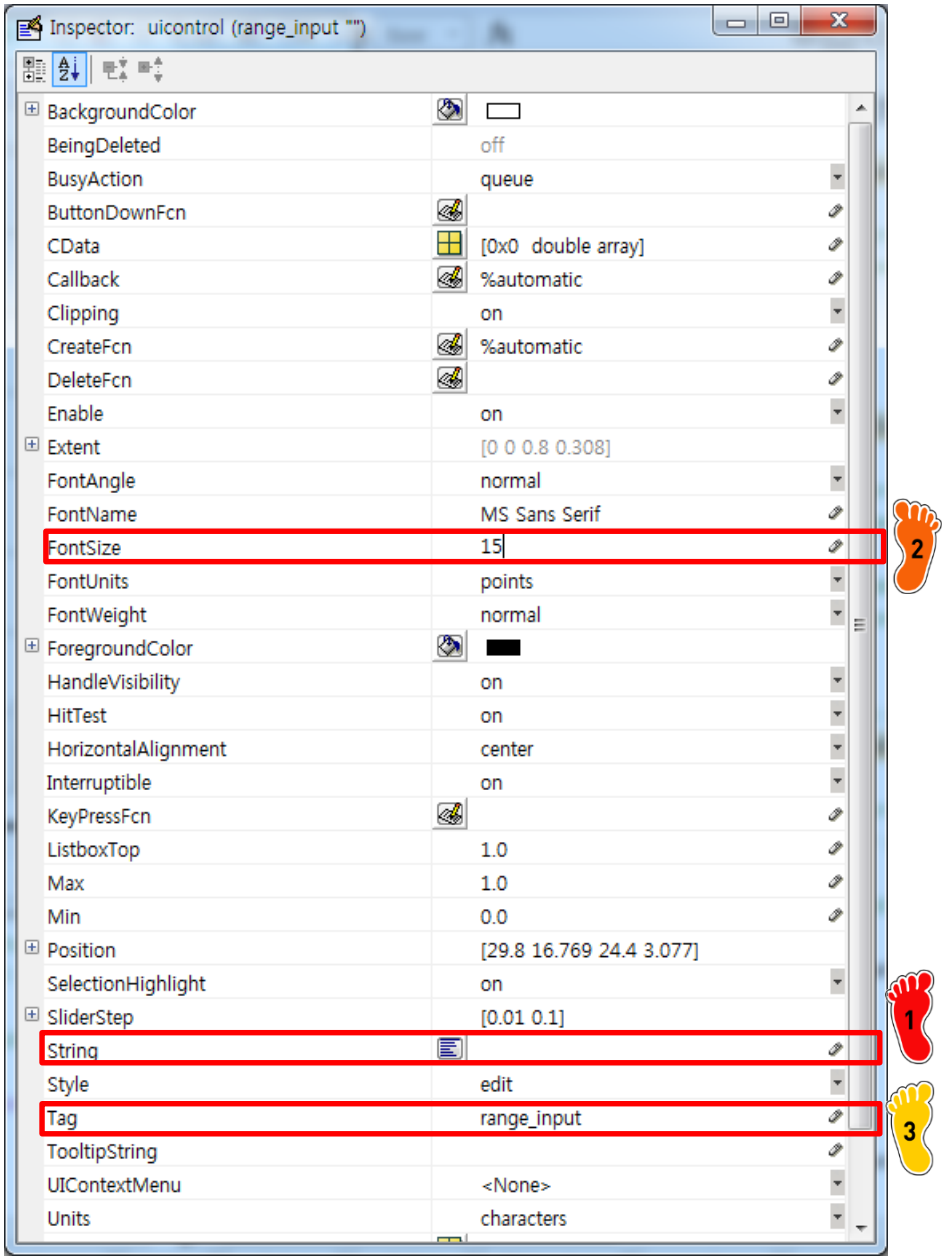
EDIT TEXT BOX



1 edit text 아이콘 클릭

2 edit text box 생성
더블 클릭

EDIT TEXT BOX: INSPECTOR



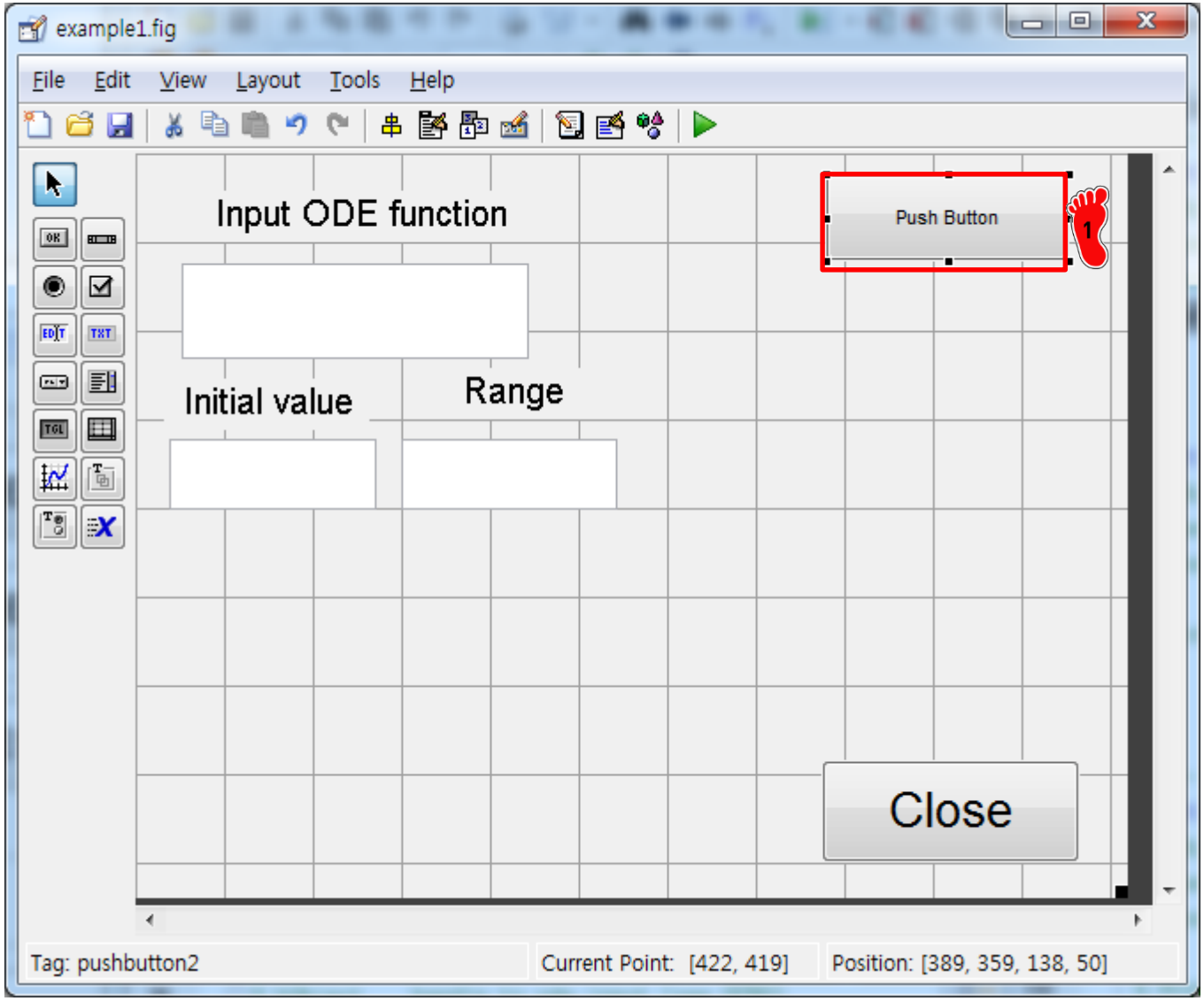
1 String 빈칸으로 변경

2 FontSize 15로 변경

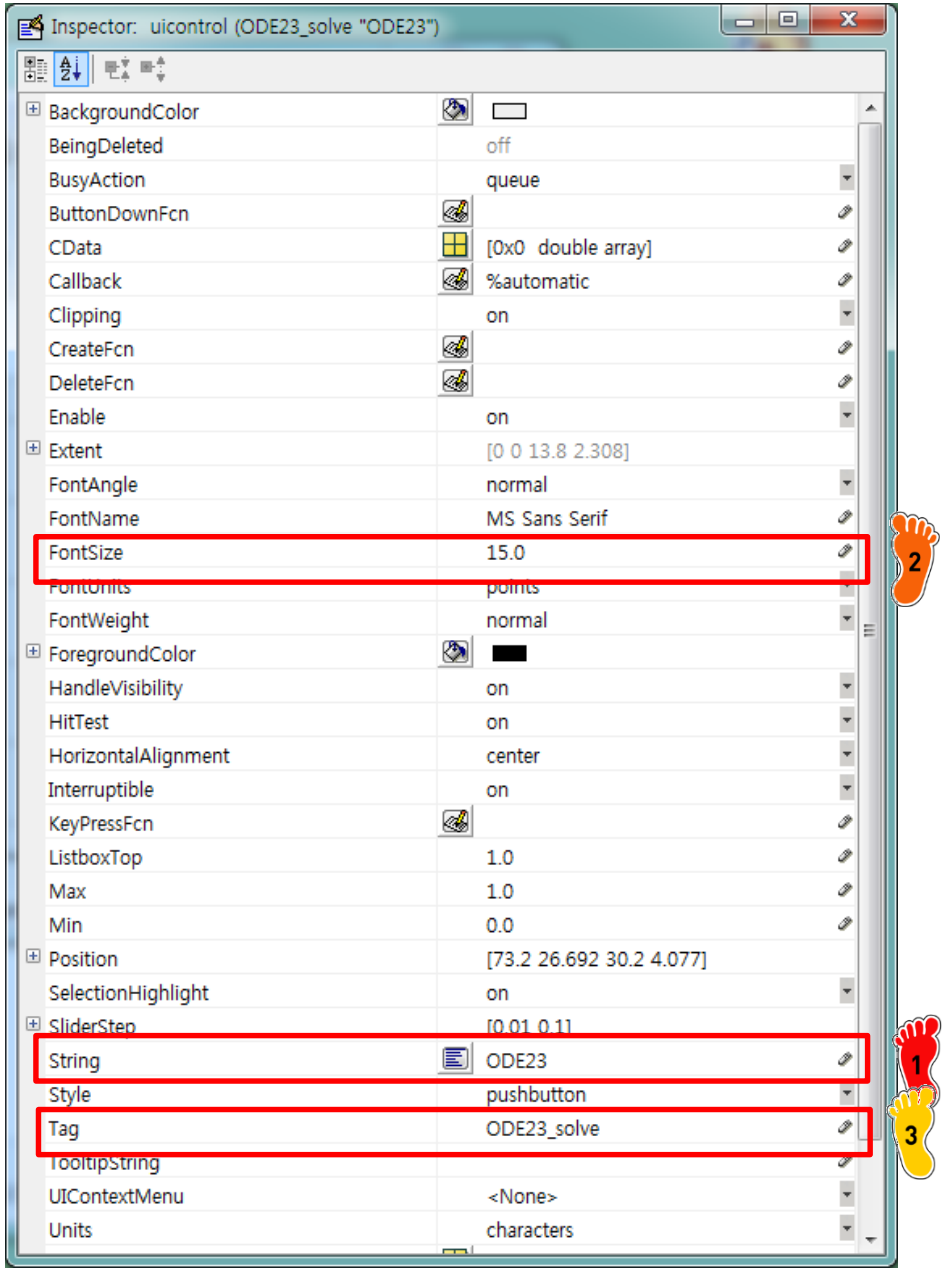
3 Tag 를 range_input 으로 변경

ODE23 PUSH BUTTON

1 pushbutton 생성



ODE23 PUSH BUTTON: INSPECTOR



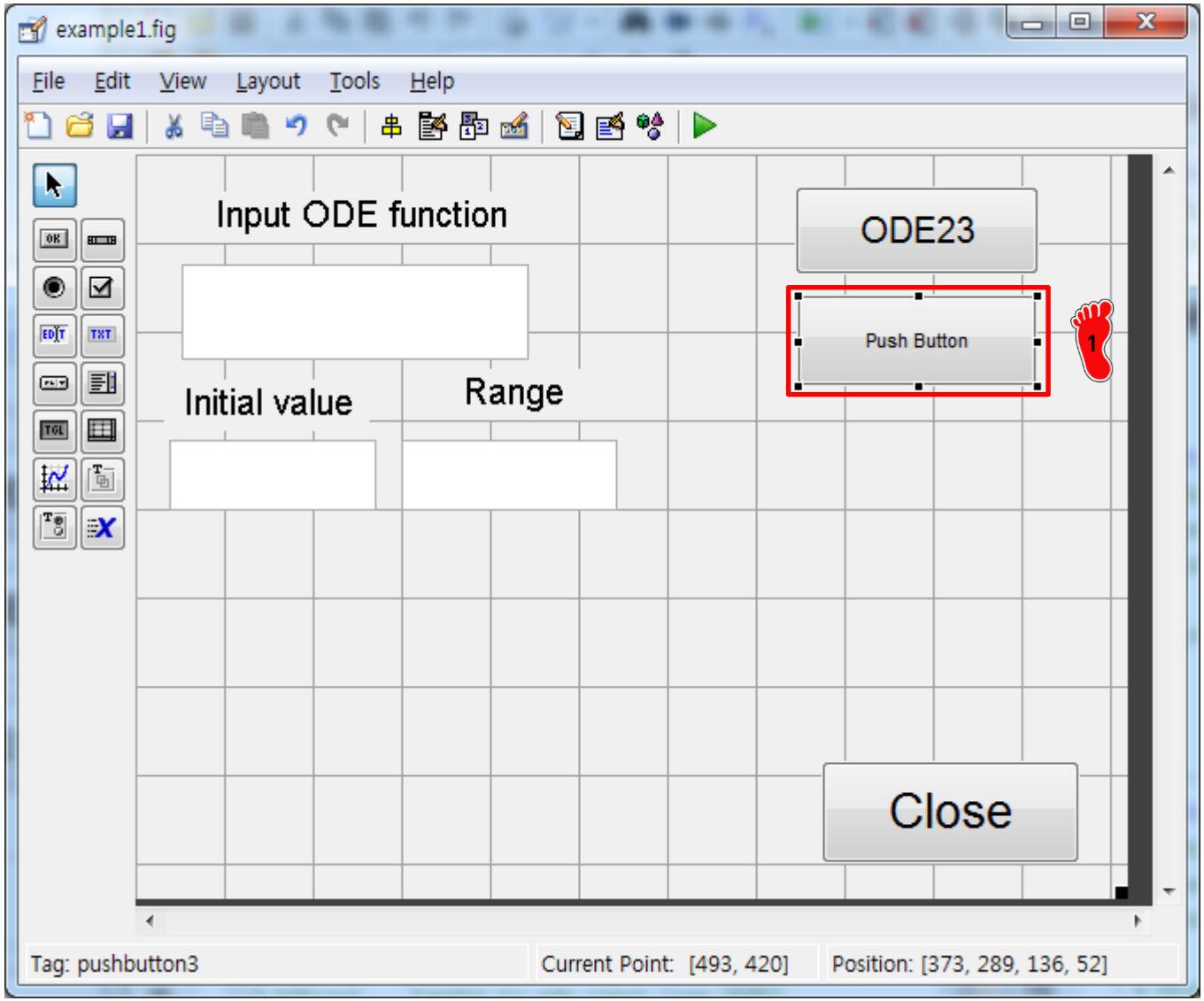
1 String ODE23으로 변경

2 FontSize 15로 변경

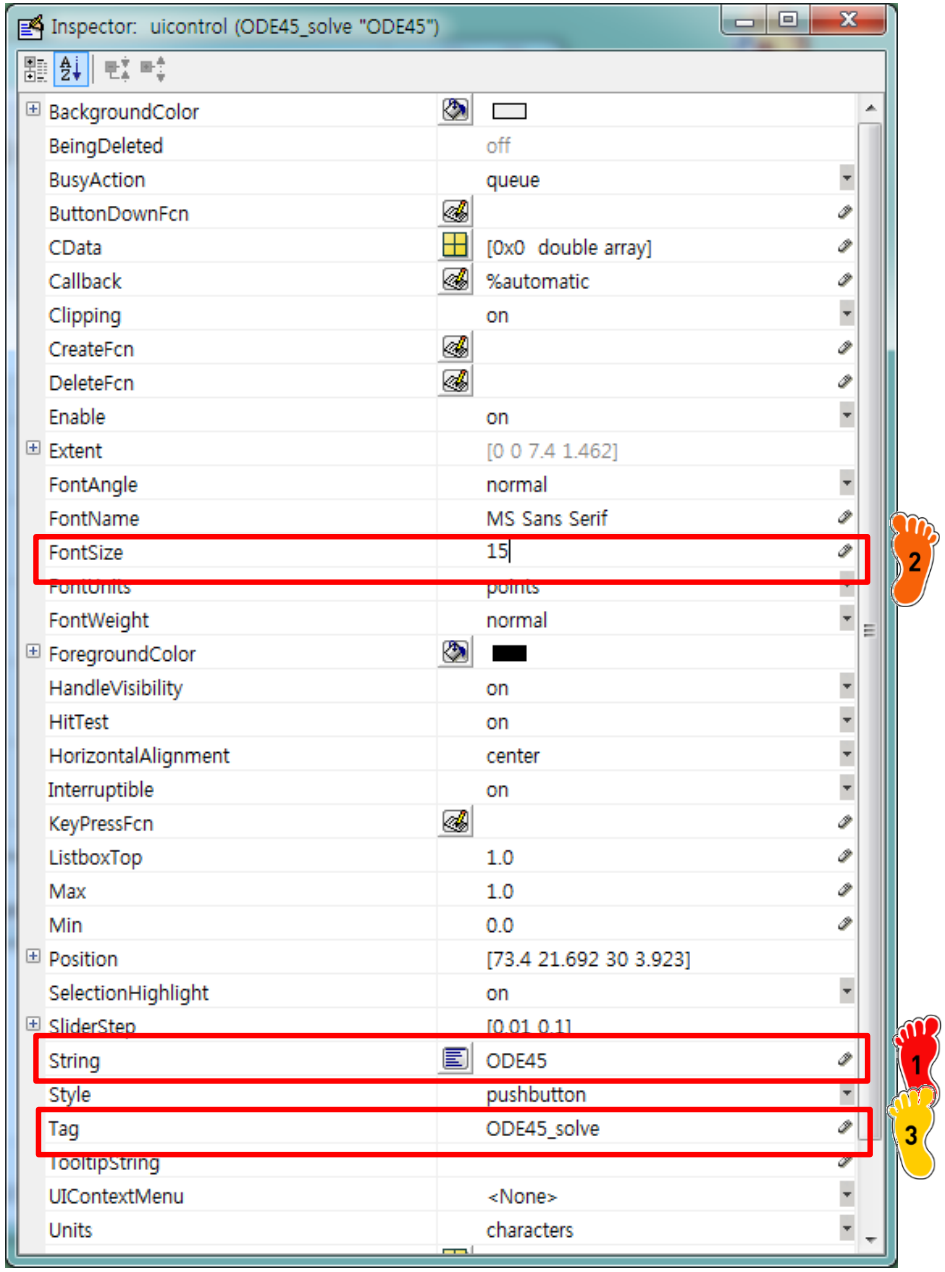
3 Tag 를 ODE23_solve 로 변경

ODE45 PUSH BUTTON

1 pushbutton 생성



ODE45 PUSH BUTTON: INSPECTOR



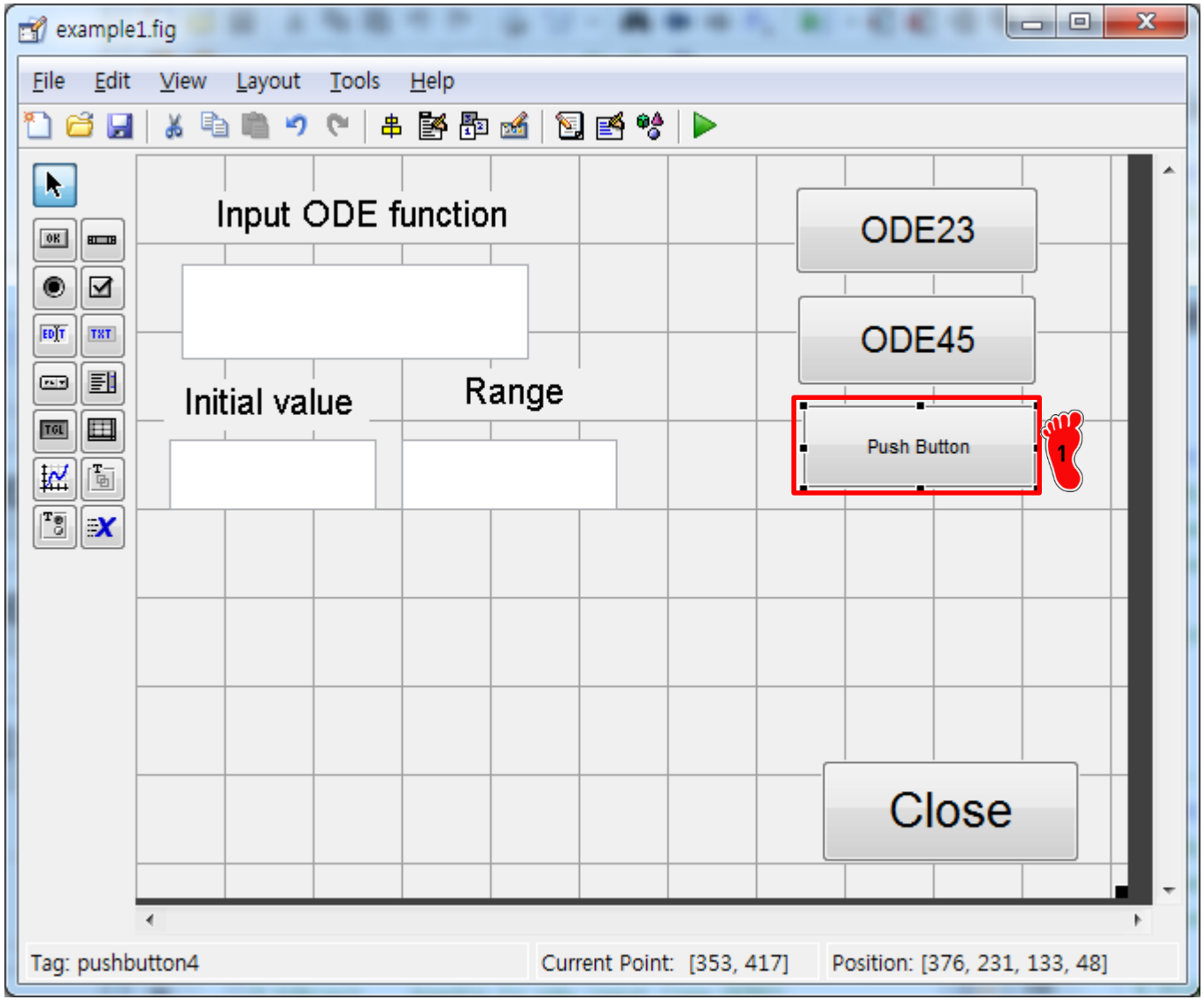
1 String ODE45 로 변경

2 FontSize 15 로 변경

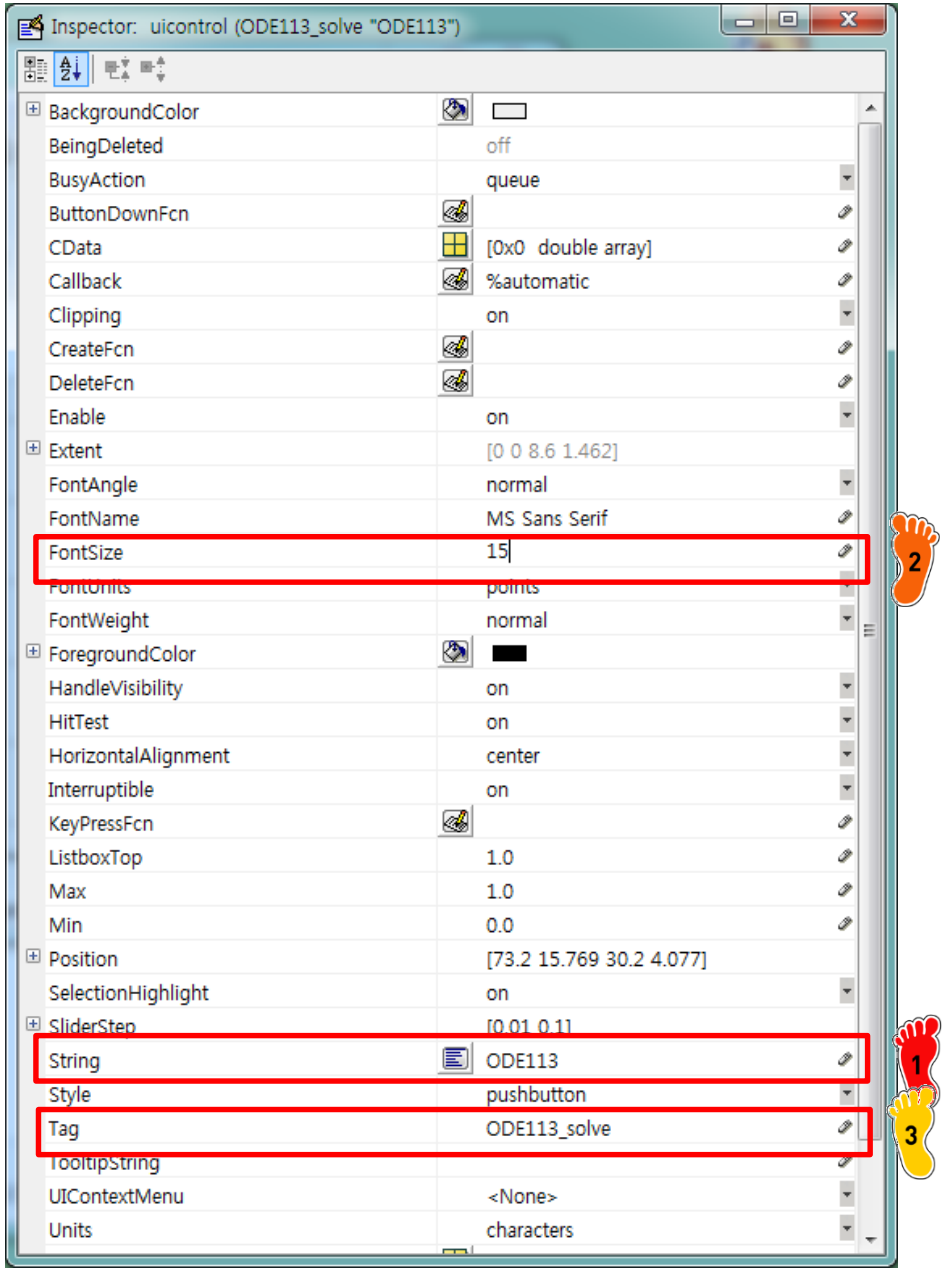
3 Tag 를 ODE45_solve 로 변경

ODE113 PUSH BUTTON

1 pushbutton 생성



ODE113 PUSH BUTTON: INSPECTOR



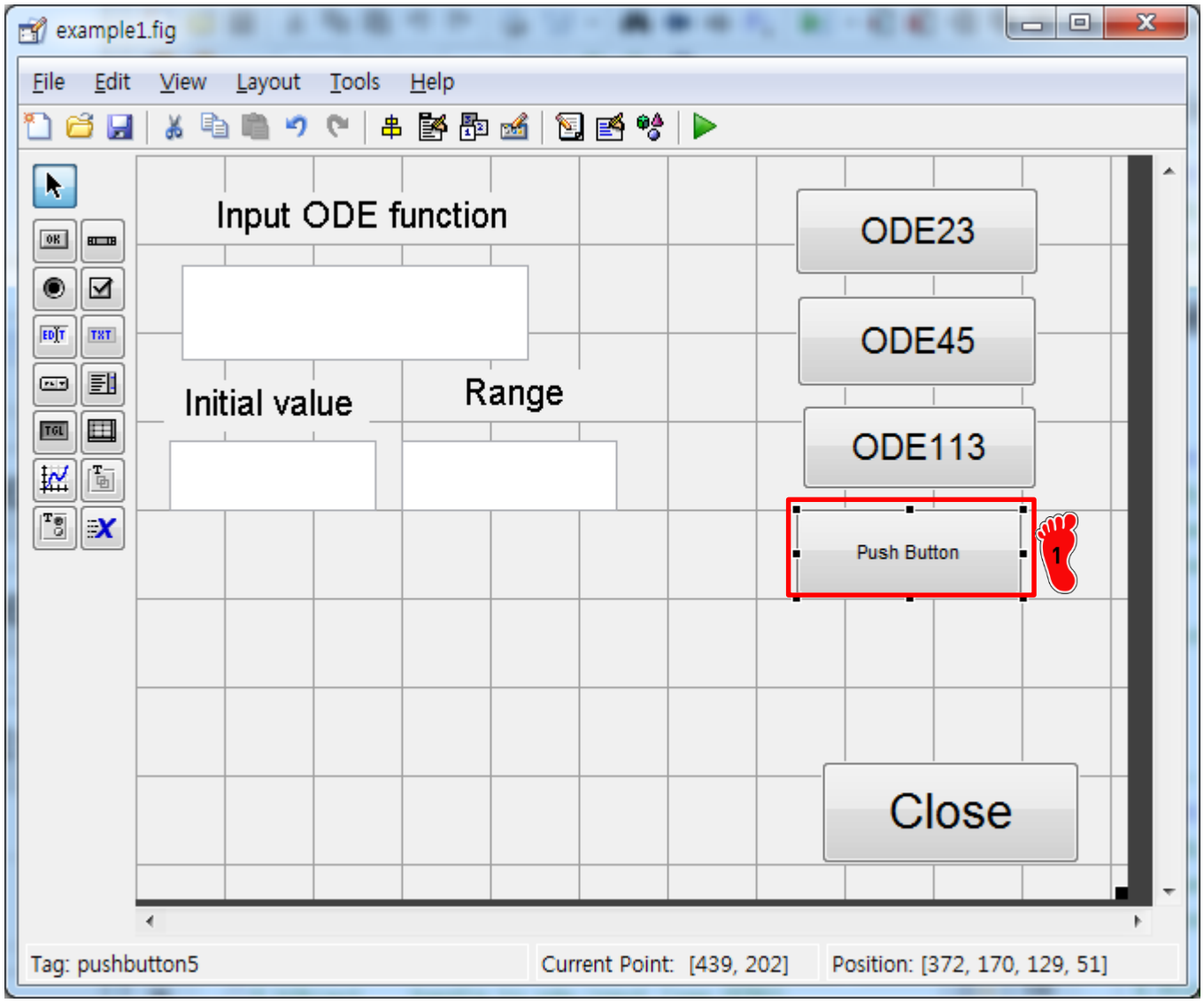
1 String ODE113 으로 변경

2 FontSize 15 로 변경

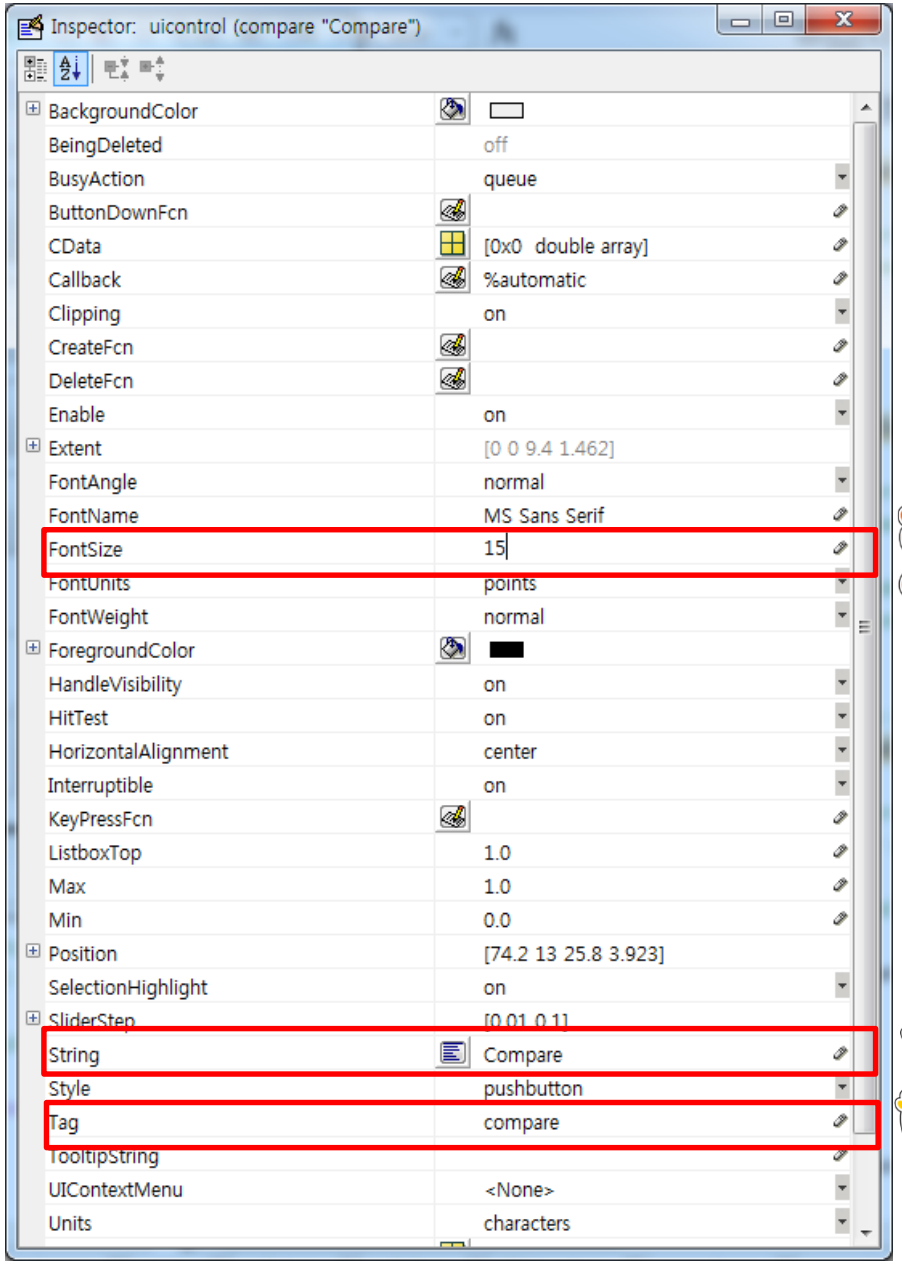
3 Tag 를 ODE113_solve 로 변경

COMPARE PUSH BUTTON

1 pushbutton 생성



COMPARE PUSH BUTTON: INSPECTO

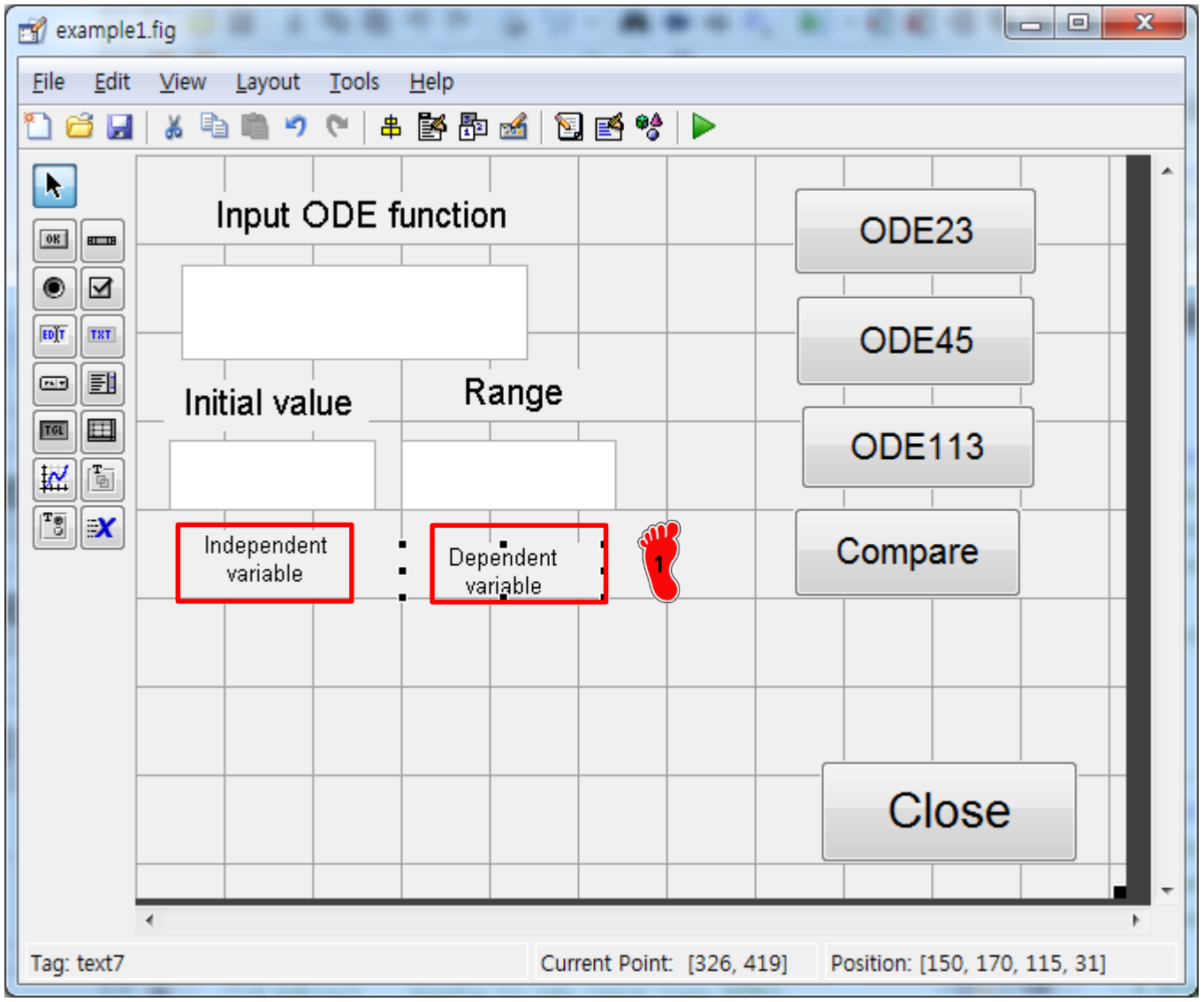


1 String Compare 으로 변경

2 FontSize 15 로 변경

3 Tag 를 compare 로 변경

STATIC TEXT BOX

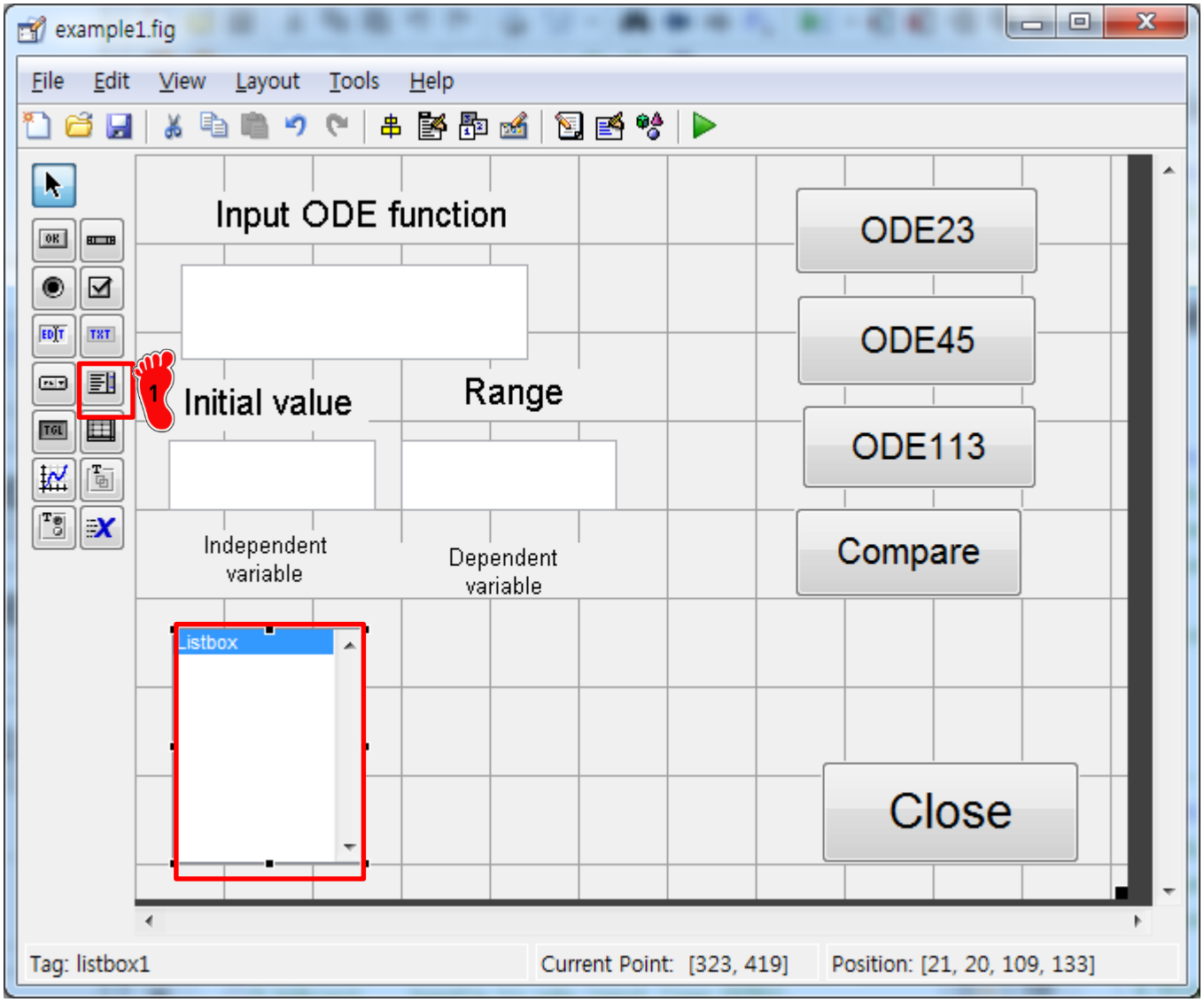


1 Independent variable 과 Dependent variable 이름을 갖는 static text box 2개 생성

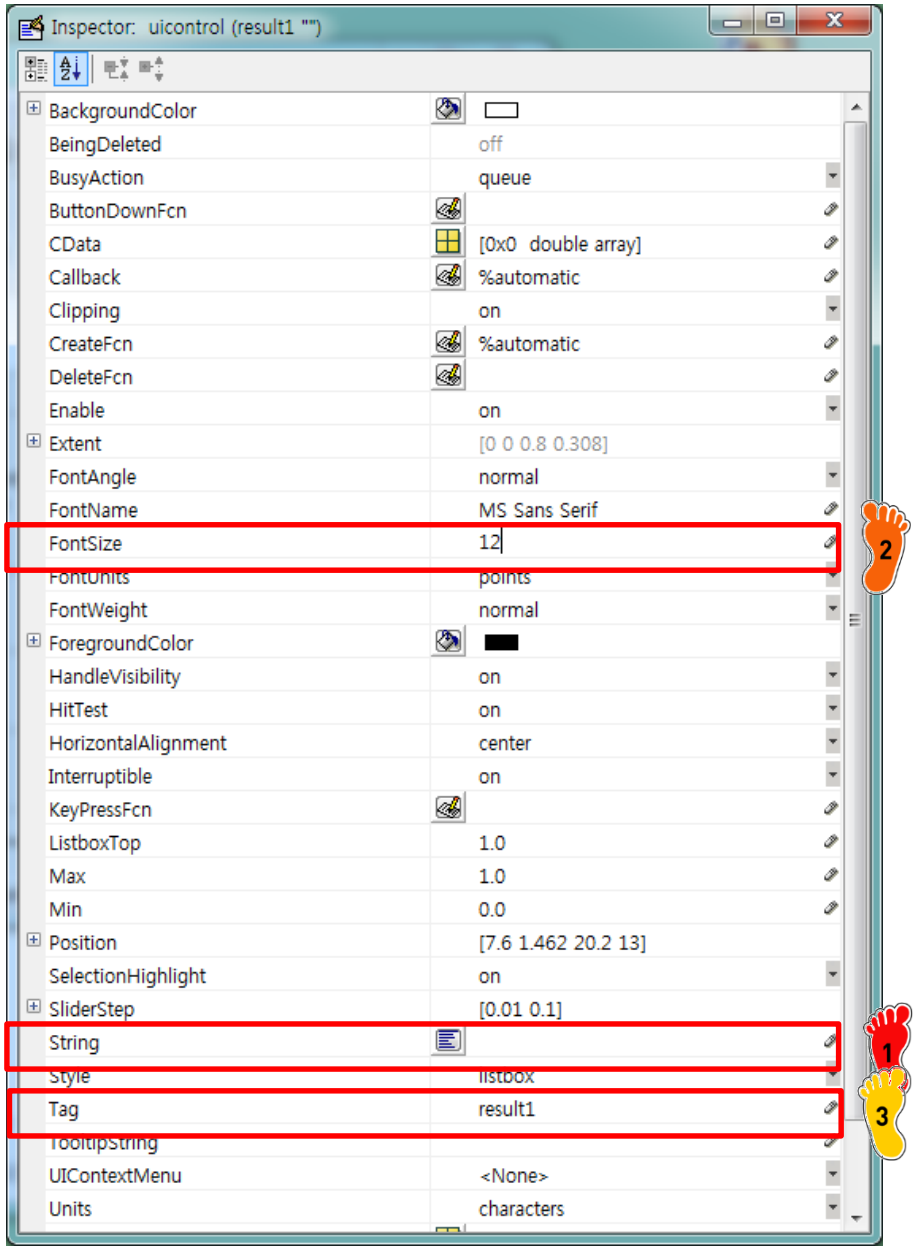
LIST BOX

1 List box icon 클릭

2 List box 생성



LIST BOX: INSPECTOR



1 String 빈칸으로 변경

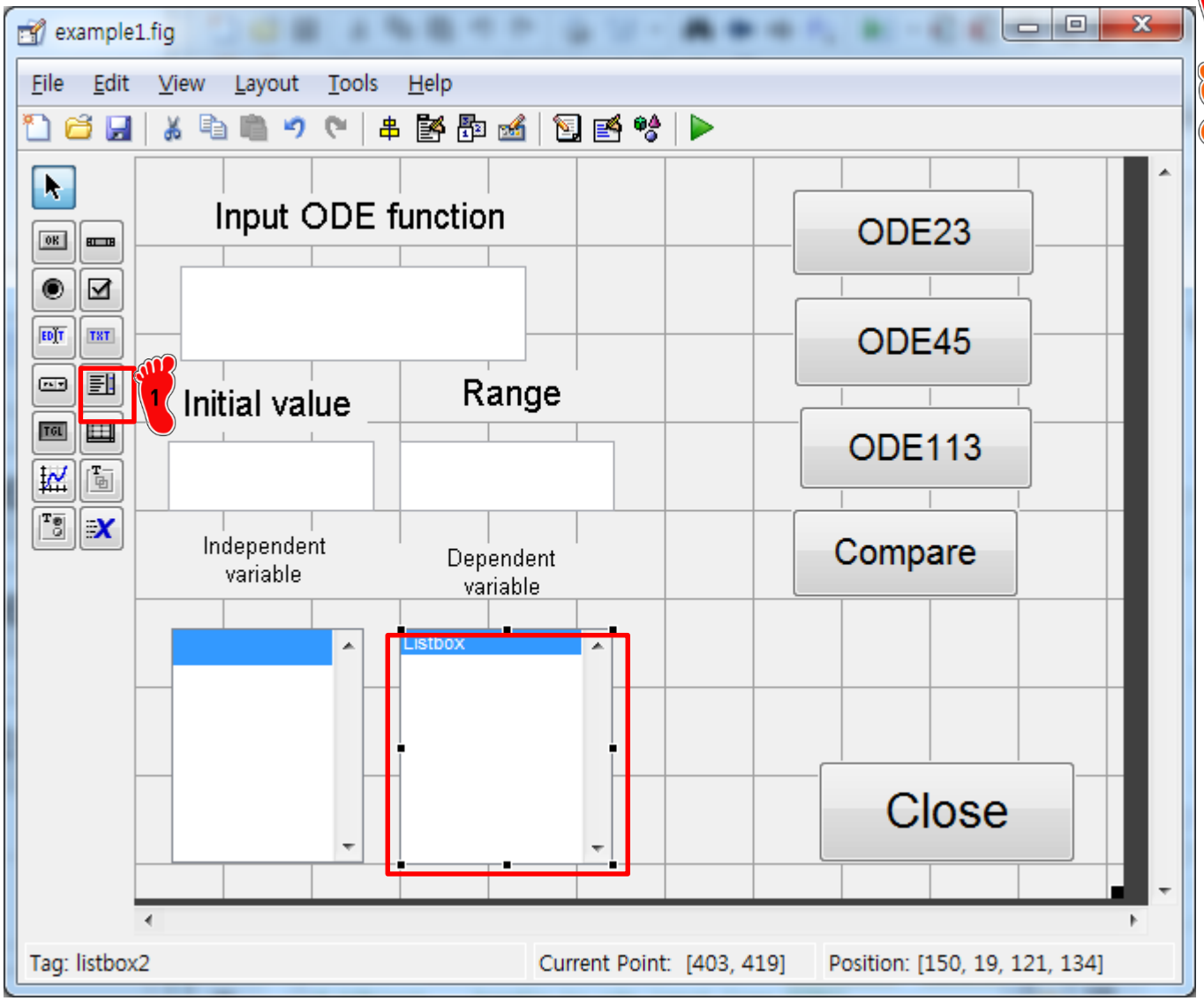
2 FontSize 12 로 변경

3 Tag 를 result1 으로 변경

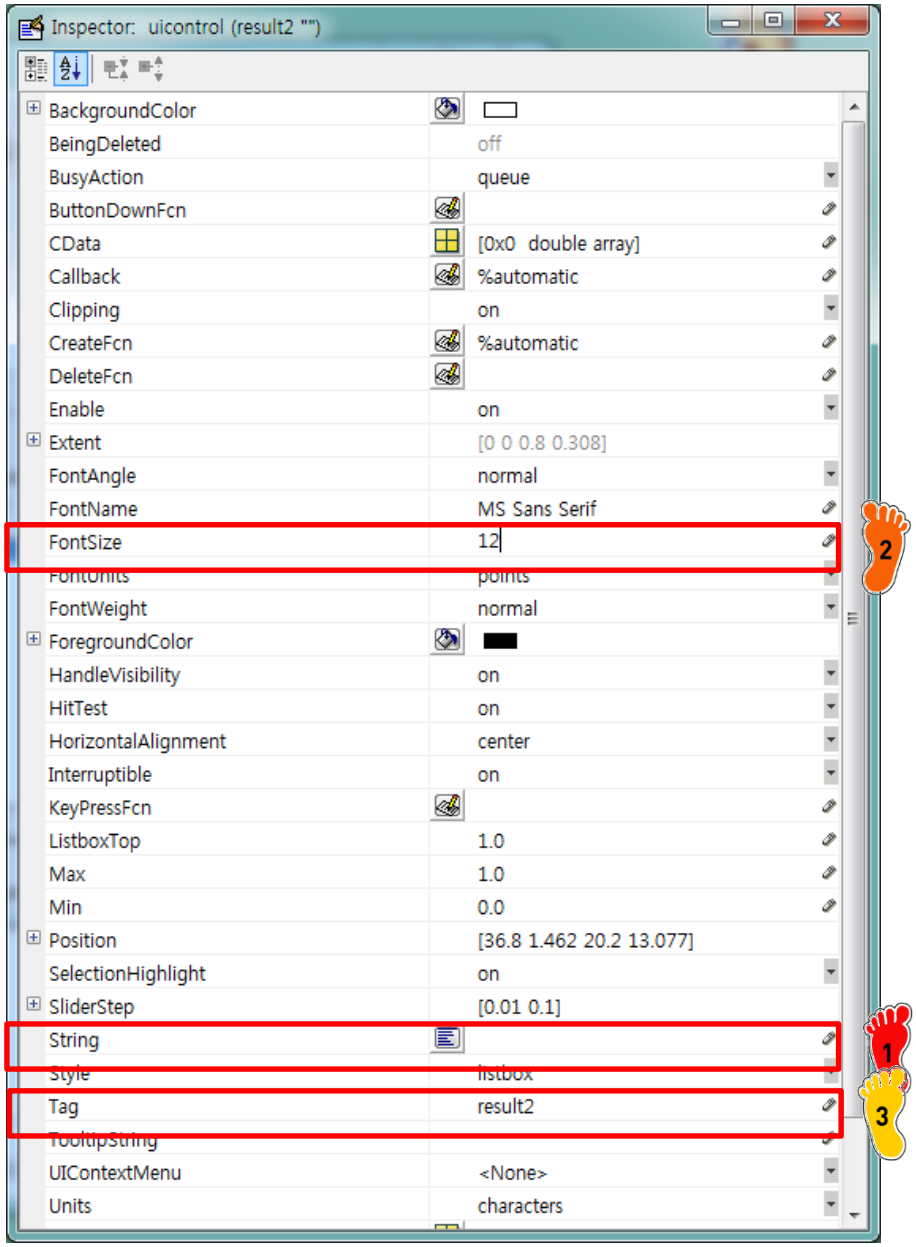
LIST BOX

1 List box icon 클릭

2 List box 생성



LIST BOX: INSPECTOR

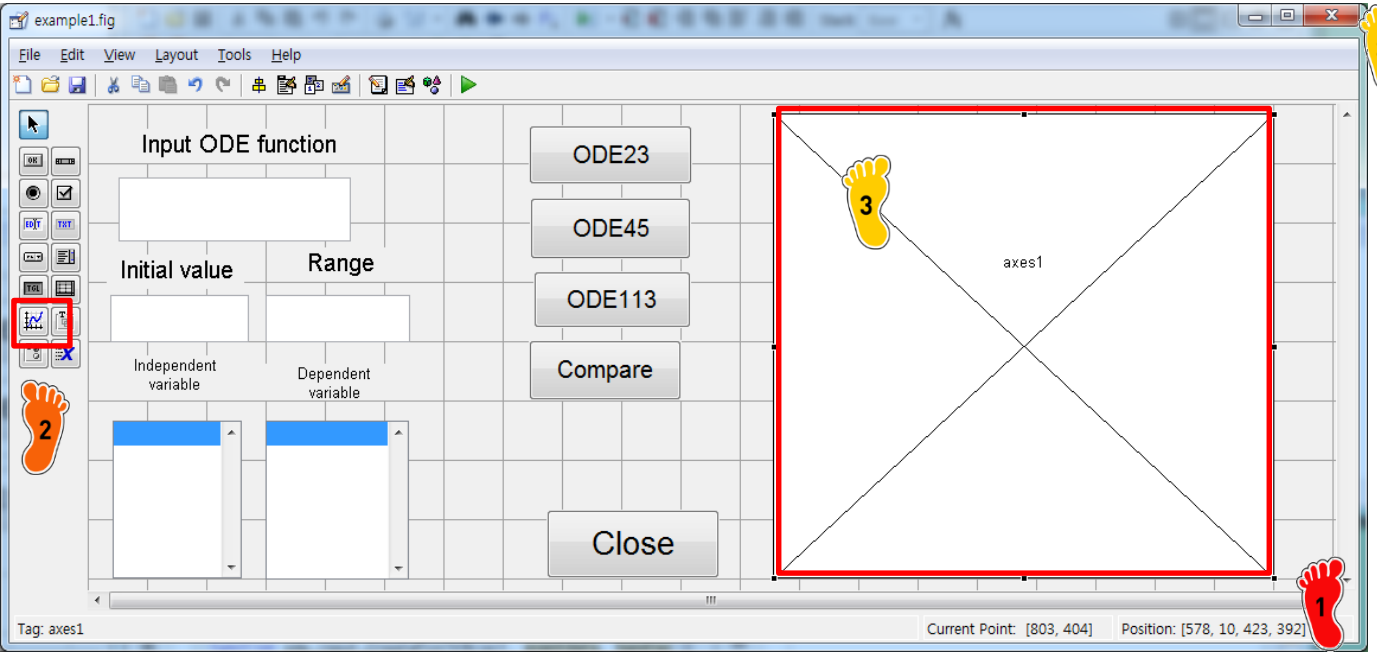


1 String 빈칸으로 변경

2 FontSize 12 로 변경

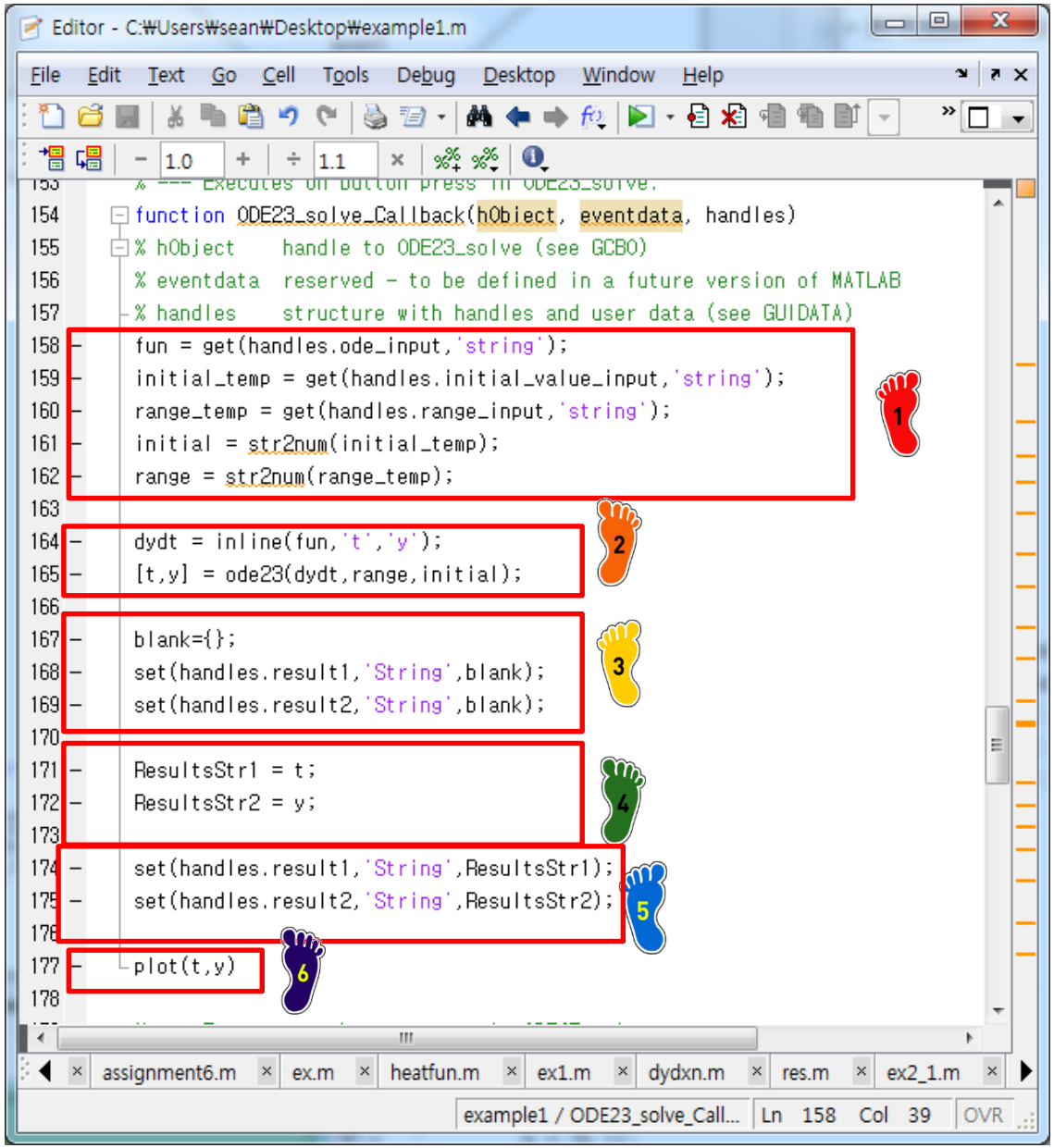
3 Tag 를 result2 로 변경

GRAPH WINDOW



1. GUI window 창을 마우스 드래그 하여 늘림
2. axes 아이콘 클릭
3. graph 를 표시해주는 창 생성

ODE23 PUSHBUTTON CODING



1. ODE23_solve 태그로 이동

   ode_input 태그로 지정된 edit box 입력 값을 get 함수로 호출

   initial 과 range 값은 숫자로 입력 해야 하기 때문에 str2num 명령어를 이용하여 숫자로 변경 후 저장

2. 호출한 함수를 inline 명령어를 이용하여 함수로 지정

   ode23 함수로 실행

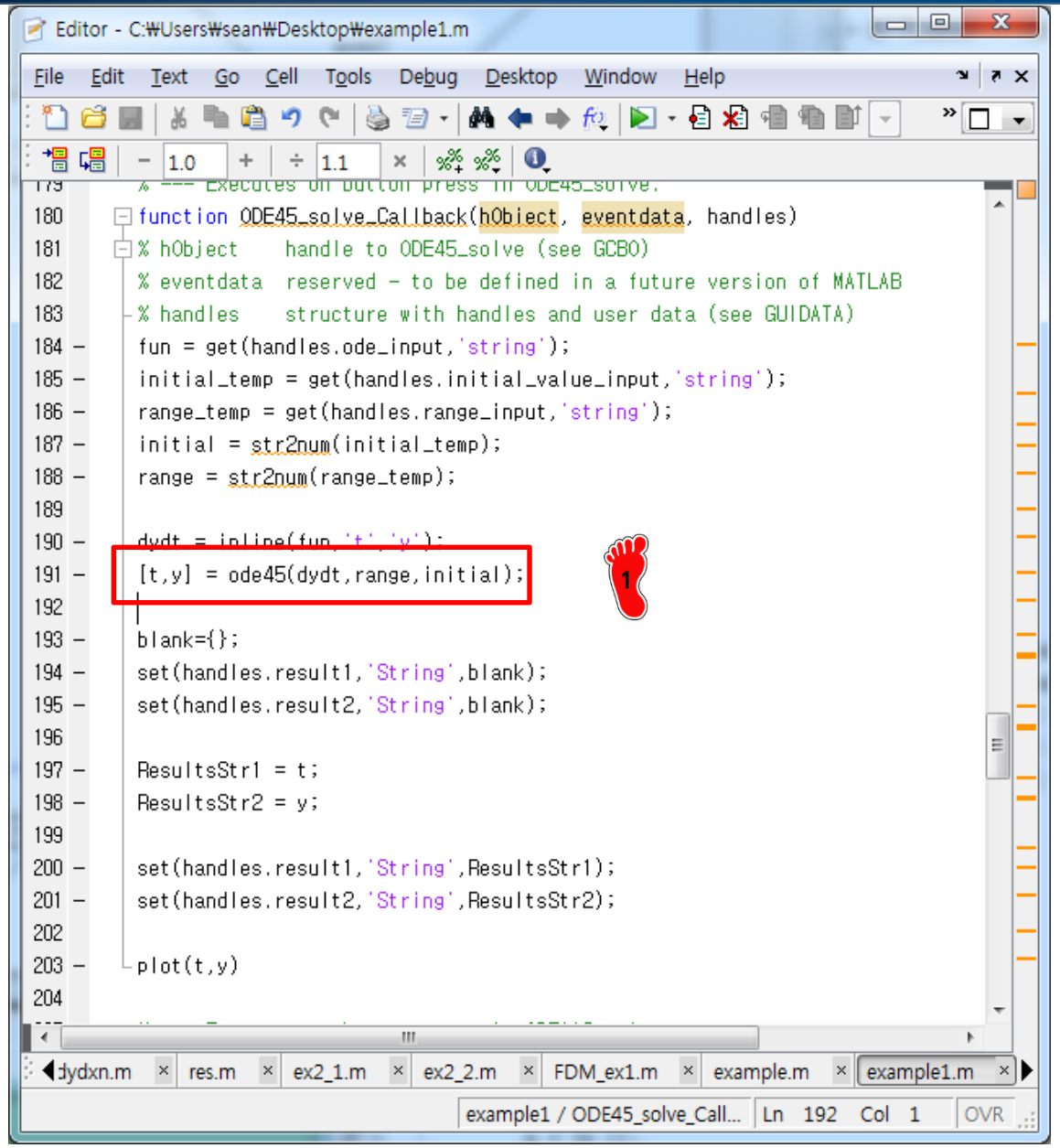
3. result1 과 result2 의 태그로 지정된 listbox 내용을 clear

4. 결과 저장

5. 결과를 listbox에 출력

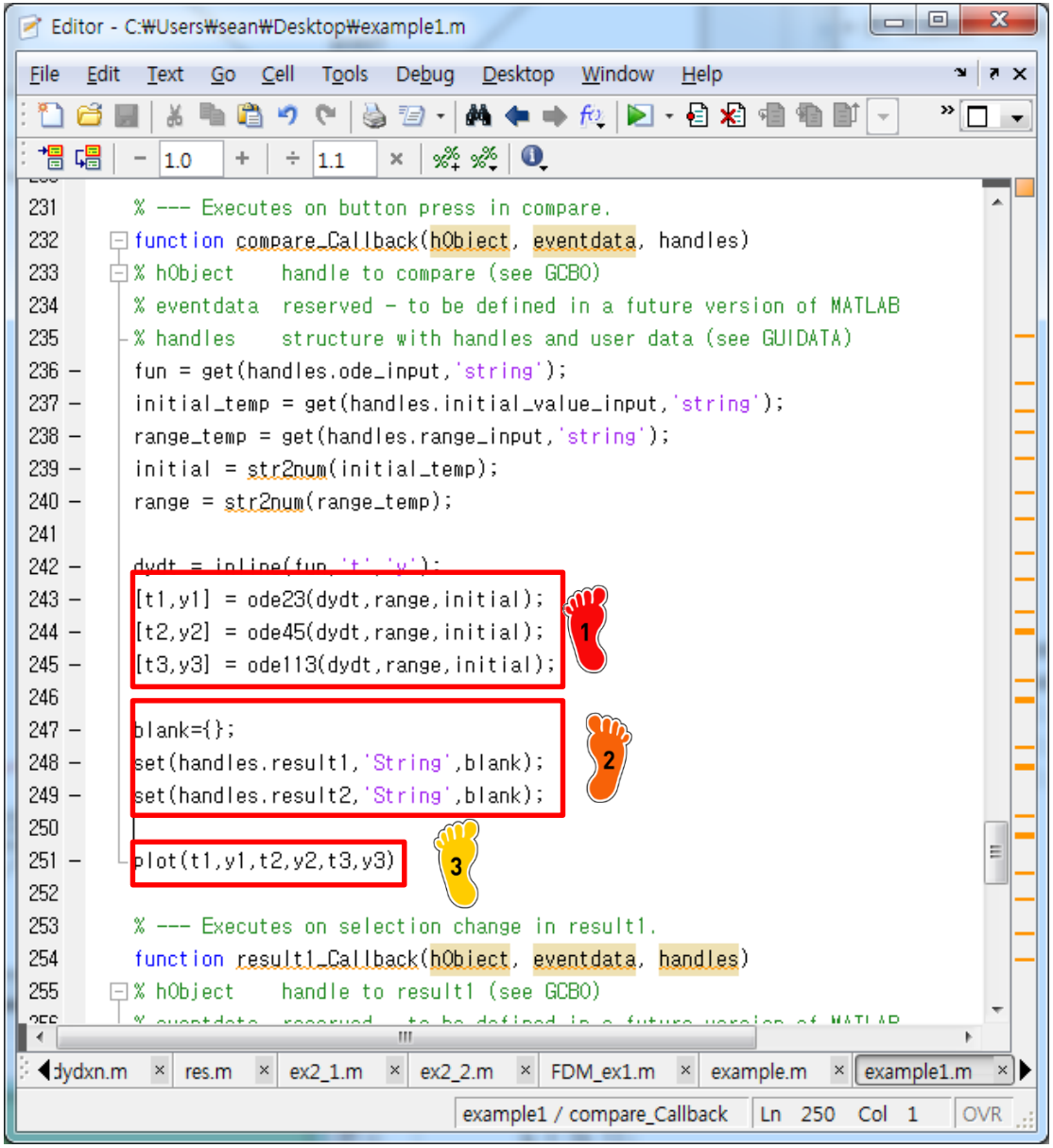
6. 그래프 출력

ODE45&113 PUSHBUTTON CODING



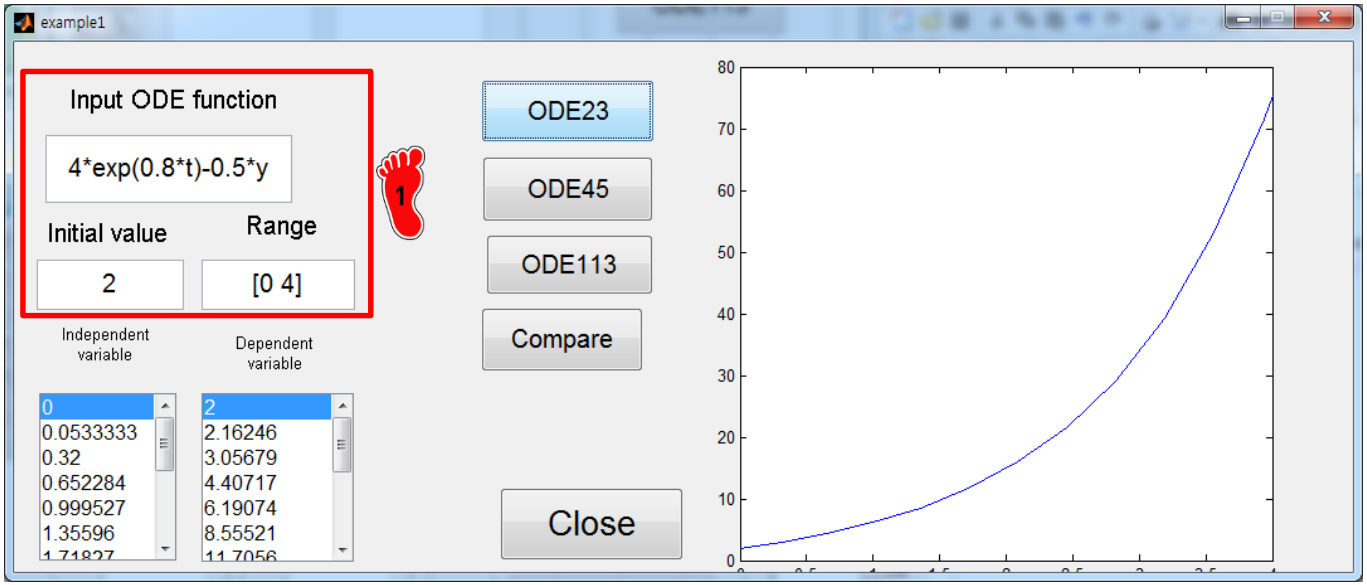
1 ode함수 명령어만 변경 후 나머지는 동일한 코드로 입력

ODE45&113 PUSHBUTTON CODING



1. 세 가지 ode 함수를 비교하기 위해 결과를 따로 저장
2. listbox 내용 출력코드는 삭제
3. 세 가지 결과를 동시에 plot

RESULT



1 함수 및 초기값과 range 입력 후 실행 버튼을 클릭하여 결과 확인

$$\frac{dy}{dx} = f(x,y) = 4e^{0.8x} - 0.5y$$

with $y(0) = 2$ from $x = 0 \sim 4$